



JAVA

PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO



Autores:

Dr. Juan Carlos García Castro
Dr. Wilson Torres Delgado
Dr. Ángel Cárdenas García
Dr. Jorge Damián Valverde Iparraguirre



Java™

JAVA

PSEUDOCÓDIGO Y
DIAGRAMAS DE FLUJO

Dr. Juan Carlos García Castro
Dr. Wilson Torres Delgado
Dr. Ángel Cárdenas García
Dr. Jorge Damián Valverde Iparraguirre



Universidad Nacional de San Martín
Vicerectorado de Investigación

Fondo Editorial

JAVA

Pseudocódigo y diagramas de flujo

Texto editado por:

Copyright © Universidad Nacional de San Martín

Vicerectorado de Investigación

Fondo Editorial

Jr. Maynas N° 177, Tarapoto, Perú

fondoeditorial@unsm.edu.pe

Autores y revisores:

Dr. Juan Carlos García Castro

Dr. Wilson Torres Delgado

Dr. Ángel Cárdenas García

Dr. Jorge Damián Valverde Iparraguirre

Diseño de portada y diagramación:

Manuel Pérez Cerquera - Eureka Studio

Derechos de autor reservados

Primera edición digital, septiembre 2023

Libro electrónico disponible en:

<https://repositorio.unsm.edu.pe/handle/11458/5409>

<http://hdl.handle.net/11458/5409>

<https://doi.org/10.51252/11458/5409>

Hecho el Depósito Legal en la Biblioteca Nacional del Perú N° 202308797

ISBN: 978-612-4356-12-4

Ninguna parte de esta publicación, incluido el diseño de cubierta, puede ser re-producida, almacenada o transmitida de modo alguna, ni por ningún medio, ya sea electrónico, químico, mecánico, óptico, de grabación o de fotocopia, sin permiso previo de los autores.

Siempre nos ha llamado la atención la diversidad de los algoritmos, y es que para cada problema existe más de una forma de resolverlos; sin embargo, también existen problemas complejos que requieren del uso de algoritmos especiales, que son funciones matemáticas o métodos adaptivos capaces de ayudarnos a resolver problemas de búsqueda complejos y optimización. Aquí en este libro nos limitaremos a solucionar problemas computacionales a través del uso de algoritmos simples.

A medida que usted desarrolle problemas usando algoritmos, se verán en la capacidad de crear nuevos problemas, dado que su cerebro está en constante entrenamiento y su lógica se vuelve más fluida, como dice un refrán alemán “la práctica hace al maestro”. La forma más común de medir cuan óptimo es un algoritmo es a través del tiempo de compilación (tiempo en llegar a la solución) y el número de pasos usados (incluye procesos y variables usadas).

Finalmente, esperamos que este libro **“JAVA: Pseudocódigo y diagramas de flujo”** se constituya en un elemento de guía y consulta para toda esa juventud universitaria estudiosa, y en la medida que ello ocurra, nuestro esfuerzo realizado se verá plenamente recompensado.

“No es el límite el infinito, si no, los límites que uno mismo nos ponemos en el camino”

Los autores

PARA OFELIA MI MADRE QUERIDA, PARA LIZ P. MI
COMPAÑERA DE VIDA, MIS FORTALEZAS LIZ Y BRUNO,
Y PARA JORGE Y FROY ADMIRABLES SIEMPRE.

Wilson

A MIS PADRES JUANITA Y ELISEO.

Jorge Damián

A MIS HIJOS: ADRIANO, LUCIANA Y FERNANDA.

Ángel

A MI ESPOSA E HIJOS Y A MIS PADRES HORACIO Y ANA.

Juan Carlos

Índice

Introducción	pág. 7
Bloque 1: Introducción y Entrada-Salida.....	pág. 16
Bloque 2: Expresiones y Operadores.....	pág. 27
Bloque 3: Estructuras Condicionales.....	pág. 51
Bloque 4: Estructuras Repetitivas.....	pág. 92
Bloque 5: Arreglos o Vectores.....	pág. 136
Bloque 6: Matrices o Tablas.....	pág. 159
Bloque 7: Cadenas.....	pág. 186
Bloque 8: Estructuras.....	pág. 210
Bloque 9: Funciones.....	pág. 244
Bloque 10: Pilas.....	pág. 309
Bloque 11: Colas.....	pág. 321
Bloque 12: Listas.....	pág. 333
Bloque 13: Archivos.....	pág. 353
Bloque 14: Programación Orientada a Objetos (POO)	pág. 381

Introducción

Algoritmo

En nuestro día a día, buscamos las soluciones óptimas para resolver nuestros problemas. Por ejemplo, al cocinar o tomarnos un baño. Siempre ideamos una estrategia, una serie de pasos, o lo hacemos de manera espontánea. Pero ¿no es todo mejor cuando tenemos un plan listo?

Al hablar de algoritmo, nos referimos a la secuencia de pasos coherentes y con lógica que nos llevarán a la resolución de un problema. Para ello, algunos autores toman en cuenta características como la observación, planteamiento, diseño, ejecución y testeos. Pero, si nos referimos a la estructura básica del algoritmo, nos encontraremos con las famosas entradas, procesos y salidas. Esto está presente en todo, y es parte fundamental del pensamiento lógico al idear y ejecutar aplicaciones.

Veremos diferentes formas de programar, como la Programación Orientada a Objetos, o la programación secuencial. No obstante, es importante reconocer que, el desarrollo de la lógica inherente a nosotros (ya sea con diagramas de flujo o pseudocódigo) nos permitirá ver a la programación de una manera más sencilla.

Se puede observar a diario a estudiantes que mencionan que, programar en Lenguajes de Programación (valga la redundancia), es lo único necesario para desarrollar la lógica necesaria para el mundo laboral o académico. No obstante, se requiere de una base, de una madurez necesaria para que, al pasar de la planificación a la ejecución, no sea tan dificultoso el desarrollar los diferentes ejercicios.

Es en ese sentido que los algoritmos son importantes, no solo al programar, sino también al analizar el cómo resolver diferentes situaciones de la vida real. A diferencia de las matemáticas, los algoritmos emplean los diferentes símbolos de manera diferente, por ejemplo, al asignar valores a una variable con un valor dado, $x = x + 2$, que en matemáticas tiene la forma de una ecuación, y sería incorrecta desde dicho punto de vista. No obstante, en los algoritmos es válido, así que hay que tener cuidado en ese aspecto. Pero haciendo una comparación con las matemáticas, los algoritmos pueden ser desarrollados de diferentes formas, diferentes caminos que dependen del usuario que esté desarrollándolos. Lo cual posibilita un mundo de ideas y fomenta la creatividad en nosotros.

Puedes hacer un algoritmo que desarrolle x problema en 5 pasos, es decir, es eficaz. Pero, lo preferible es buscar la eficiencia del algoritmo, desarrollando una serie de pasos que permitan la solución de un problema con la mínima cantidad de recursos.

Características de un Algoritmo

Los algoritmos presentan las siguientes características:

- **Entradas:** Los algoritmos por lo general tienen de cero a más entradas. Es decir, los valores de entrada que serán utilizados para su posterior procesamiento. Visto desde otro ángulo, podríamos imaginarnos a una función en matemáticas, que requiere un valor de entrada en “x” para ser procesado, y tener su respectiva “y”. En este caso, las entradas vendrían a ser las abscisas.
- **Proceso:** Es cuando los datos de entrada son procesados, transformados o son parámetros de referencia para que, a partir de ellos, se pueda producir nueva información o datos a partir de él.
- **Salidas:** Las salidas son la consecuencia de las entradas y su posterior procesamiento. El algoritmo puede tener de cero a más salidas. Pero, lo recomendable sería que al menos tenga una salida, si no ¿Qué es lo que realmente pretendríamos al no tener un resultado?
- **Control:** Los parámetros de control se pueden hallar preguntándonos, ¿Qué datos o información se empleará en el proceso? ¿Cuándo iniciará? ¿Cuándo acabará?
- **Óptimo:** Como se mencionó anteriormente, lo preferible al desarrollar un algoritmo es que cumpla al resolver el problema, pero que también lo haga con la mínima cantidad de recursos posibles. Para que se facilite el proceso de los datos y el sistema sea más eficiente.

Metodologías para la elaboración de un algoritmo

Las metodologías serán una ayuda sencilla para desarrollar un algoritmo, ya sea de manera gráfica (con símbolos, flechas, figuras geométricas), o con palabras que utilizamos en el día a día (mostrar, imprimir, etc.).

Diagrama de flujo

Un diagrama de flujo, es una representación gráfica de la secuencia de pasos que se emplean al desarrollar un algoritmo. Es una herramienta visual para mostrar la secuencia de decisiones y acciones que tomaremos en cuenta al resolver un problema.

Para ello, se hacen uso de flechas, conectores, símbolos, y formas geométricas que representan diferentes funciones, como mostrar en pantalla, imprimir, tomar datos por teclado, periféricos, entre otros.

Al emplear un diagrama de flujo, se fomenta el desarrollo de la lógica del individuo, pues, de una manera amigable se busca visualizar de manera clara y estructurada cómo se emplea un algoritmo. De esta manera, se facilita la comprensión y análisis para tener una base clara antes de lanzarnos al mundo de la programación en diferentes lenguajes, como Java, Python, C++, entre otros.

Este diagrama de flujo, al ser una representación de un algoritmo, tiene que seguir la estructura básica del mismo. Es decir, inicio, proceso y salida. Pero ya depende del autor del algoritmo si desea hacerlo de manera secuencial, o por bloques. Cabe destacar que ambos métodos serán aplicados en el desarrollo del presente libro.

Pseudo código

A diferencia de los diagramas de flujo que representan un algoritmo usando símbolos (como figuras o flechas), el pseudo código representa el algoritmo, pero utilizando un lenguaje simplificado. Esta es una combinación del lenguaje humano y los lenguajes de programación. Pero, a diferencia de los Lenguajes de Programación, el pseudocódigo no sigue una sintaxis rigurosa, pero sirve como una herramienta útil para el diseño al describir el flujo lógico de un algoritmo.

El pseudo código permite expresar el funcionamiento de un algoritmo de manera comprensible, y describir la lógica de los algoritmos de manera estructurada y ordenada usando instrucciones y estructuras de control similares a las que se encuentran en los lenguajes de programación convencionales.

Entre sus ventajas, tenemos que puede ser entendido por personas que no necesariamente tengan conocimientos complejos o si quiera intermedios en programación, lo que facilita la comunicación y colaboración entre diferentes personas involucradas en el desarrollo del algoritmo.

Una vez desarrollado el pseudo código, podemos expresarlo o traducirlo a un lenguaje de programación. Es decir, convertir las instrucciones y estructuras de control del pseudocódigo en las sintaxis y estructuras equivalentes del lenguaje de programación elegido.

Para el presente libro, se desarrollarán los ejercicios tanto en pseudo código como en el lenguaje de programación Java. Primero, se mostrará el pseudo código, y luego la traducción, implementación o adaptación al lenguaje Java. Pues cabe recalcar que, al tener sintaxis y palabras reservadas diferentes, la traducción puede tomar varios rumbos.

Invitamos a los lectores a desarrollar sus propias traducciones con la lógica presente en el libro, o con la lógica de ustedes mismos. De manera que se pueda hacer una comparación y contraste entre los diferentes tipos de implementación de algoritmos que iremos desarrollando conforme avance el libro.

Variables

El lenguaje de programación Java es ampliamente utilizado en el desarrollo de aplicaciones y sistemas de software. Una de las características fundamentales de Java es su capacidad para trabajar con variables. Las variables son elementos esenciales en cualquier programa, ya que permiten almacenar y manipular datos durante la ejecución del mismo. En este trabajo, exploraremos en detalle el concepto de variables en Java, su declaración, tipos, inicialización, alcance y convenciones de nombres. A lo largo de las siguientes páginas, obtendremos un conocimiento profundo sobre cómo trabajar con variables en Java y su importancia en el desarrollo de aplicaciones.

Declaración de Variables

En Java, las variables deben ser declaradas antes de ser utilizadas. La declaración de una variable incluye su tipo y su nombre. Java es un lenguaje de programación de tipado estático, lo que significa que cada variable debe tener un tipo específico que determina el rango de valores y las operaciones que se pueden realizar con ella. Algunos de los tipos de variables más comunes en Java son:

- int: para representar números enteros.
- double: para representar números decimales.
- boolean: para representar valores de verdadero o falso.
- String: para representar cadenas de caracteres.

Además de estos tipos básicos, Java también permite la creación de variables de tipos personalizados mediante la definición de clases.

Inicialización de variables

Cuando se declara una variable en Java, también se puede asignar un valor inicial en el mismo momento. A esto se le conoce como inicialización de variables. La inicialización puede realizarse utilizando un valor constante o mediante una expresión más compleja. Es importante destacar que las variables deben ser inicializadas antes de ser utilizadas en cualquier operación.

Alcance de las variables

El alcance de una variable se refiere a la parte del programa en la que la variable es visible y puede ser utilizada. En Java, el alcance de una variable puede ser local o global. Las variables locales son declaradas dentro de un bloque de código, como un método o una estructura de control, y solo son visibles dentro de ese bloque. Por otro lado, las variables globales son declaradas fuera de cualquier método o estructura de control y son accesibles desde cualquier parte de la clase.

Constantes

En Java, se pueden declarar variables que tienen un valor constante, es decir, un valor que no puede cambiar después de haber sido asignado. A estas variables se les llama constantes y se declaran utilizando la palabra clave "final". Las constantes son útiles para representar valores fijos que se utilizan en el programa y que no deben ser modificados.

Convenciones de nombres

En Java, se siguen convenciones de nombres para variables. Estas convenciones ayudan a que el código sea más legible y comprensible. Algunas pautas comunes incluyen utilizar nombres descriptivos para las variables, comenzar los nombres con letras minúsculas y seguir el estilo de escritura de camelCase. Para las constantes, se utilizan nombres en mayúsculas y se separan las palabras con guiones bajos.



Introducción a los Bloques y Concretos básicos

Bloque 1: Introducción y Entrada-Salida

En este bloque, las variables están relacionadas con la interacción del programa con el usuario y la manipulación de datos de entrada y salida. Algunas variables comunes en este contexto son:

- o Variables de tipo String: se utilizan para almacenar cadenas de caracteres que representan mensajes o datos ingresados por el usuario.
- o Variables de tipo int, double, u otros tipos numéricos: se utilizan para almacenar valores numéricos ingresados por el usuario o generados durante la ejecución del programa.
- o Variables de tipo Scanner: se utilizan para leer datos de entrada ingresados por el usuario desde la consola.

Bloque 2: Expresiones y Operadores

En este bloque, las variables se utilizan para realizar operaciones matemáticas y lógicas. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `int`, `double` o `float`: se utilizan para almacenar los resultados de operaciones aritméticas como suma, resta, multiplicación o división.
- Variables de tipo `boolean`: se utilizan para almacenar resultados de expresiones lógicas, como comparaciones o condiciones.

Bloque 3: Estructuras condicionales

En este bloque, las variables se utilizan para evaluar condiciones y tomar decisiones en base a ellas. Las variables comunes en este contexto son:

- Variables de tipo `boolean`: se utilizan para almacenar el resultado de expresiones lógicas que determinan si se cumple una condición o no.
- Variables de tipo `int`, `double` o `float`: se utilizan para almacenar valores que se comparan en las estructuras condicionales, como en las sentencias `if-else`.

Bloque 4: Estructuras repetitivas

En este bloque, las variables se utilizan para controlar la ejecución repetida de un conjunto de instrucciones. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `int`: se utilizan como contador o índice para controlar la repetición de un bucle `for` o `while`.
- Variables de tipo `boolean`: se utilizan para determinar si se cumple o no una condición de salida de un bucle.

Bloque 5: Arreglos o vectores

En este bloque, las variables se utilizan para almacenar conjuntos de valores del mismo tipo. En Java, los arreglos o vectores son estructuras de datos que permiten almacenar múltiples valores en una sola variable. Por ejemplo:

- Variables de tipo `int[]`, `double[]` o `String[]`: se utilizan para almacenar múltiples valores de un tipo específico en un arreglo.

Bloque 6: Matrices o tablas

En este bloque, las variables se utilizan para almacenar datos en una estructura de matriz o tabla bidimensional. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `int[][]`, `double[][]` o `String[][]`: se utilizan para almacenar datos en una estructura de matriz bidimensional.

Bloque 7: Cadenas

En este bloque, las variables se utilizan para trabajar con cadenas de caracteres. En Java, las cadenas se representan mediante la clase `String`. Algunas variables comunes en este contexto son:

Variables de tipo `String`: se utilizan para almacenar y manipular cadenas de caracteres, como nombres, direcciones o mensajes.

Bloque 8: Estructuras

En este bloque, las variables se utilizan para almacenar datos en estructuras de datos más complejas, como listas, pilas, colas, etc. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `ArrayList`, `LinkedList` u otras clases de estructuras de datos: se utilizan para almacenar y manipular conjuntos de elementos en una estructura específica.

Bloque 9: Funciones

En este bloque, las variables se utilizan para pasar valores entre funciones y almacenar resultados temporales. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `int`, `double` o cualquier otro tipo necesario para almacenar resultados de una función.
- Variables de tipo parámetro: se utilizan para recibir valores de entrada en una función.

Bloque 10: Pilas

En este bloque, las variables se utilizan para implementar estructuras de datos tipo pila. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `Stack`: se utilizan para almacenar y manipular elementos en una estructura de pila.

Bloque 11: Colas

En este bloque, las variables se utilizan para implementar estructuras de datos tipo cola. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `Queue`: se utilizan para almacenar y manipular elementos en una estructura de cola.

Bloque 12: Listas

En este bloque, las variables se utilizan para implementar estructuras de datos tipo lista. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `ArrayList`, `LinkedList` u otras clases de estructuras de datos: se utilizan para almacenar y manipular conjuntos de elementos en una estructura de lista.

Bloque 13: Archivos

En este bloque, las variables se utilizan para trabajar con archivos, tanto para leer como para escribir datos en ellos. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo `File`: se utilizan para representar archivos en el sistema de archivos.
- Variables de tipo `Scanner` o `BufferedReader`: se utilizan para leer datos de un archivo.
- Variables de tipo `FileWriter` o `BufferedWriter`: se utilizan para escribir datos en un archivo.

Bloque 14: POO

En este bloque, las variables se utilizan para representar objetos y almacenar datos relacionados con ellos. Algunos tipos de variables comunes en este contexto son:

- Variables de tipo clase: se utilizan para representar objetos de una clase específica.
- Variables de tipo instancia: se utilizan para almacenar datos específicos de una instancia u objeto en particular.

Recuerda que estos son solo ejemplos de variables que se utilizan comúnmente en cada uno de los bloques mencionados. La elección y uso de variables dependerá del contexto y los requisitos específicos de cada programa o aplicación que estés desarrollando en Java.

Bloque 01: Entrada y Salida

1.1.

Escribe un programa que lea de la entrada estándar dos números y muestre en la salida estándar su suma, resta, multiplicación y división.

1.1.1. Código Pseint

Algoritmo Ejercicio 1

Escribir “Ingresar primer número”

Leer nro1

Escribir “Ingresar segundo número”

Leer nro2

suma = nro1 + nro2

resta = nro1 - nro2

multiplicacion = nro1*nro2

division = nro1/nro2

Escribir “La suma es:”, suma

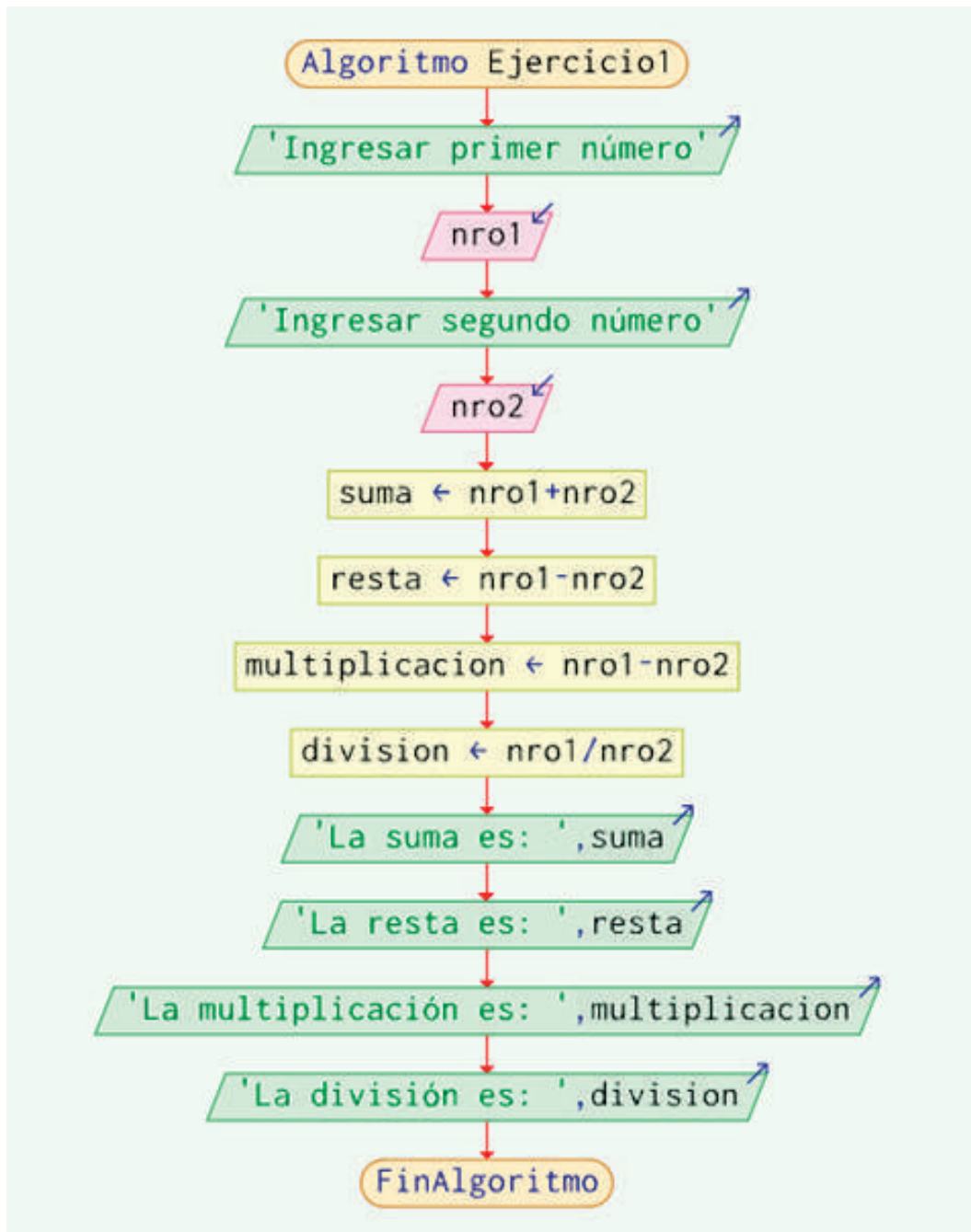
Escribir “La resta es:”, resta

Escribir “La multiplicación es:”, multiplicacion

Escribir “La división es:”, division

FinAlgoritmo

1.1.2. Diagrama de flujo



1.1.3. Adaptación a Java

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) {
        Scanner valor = new Scanner(System.in);
        double numero1, numero2, division;

        System.out.println("Insertar primer #:");
        numero1 = valor.nextDouble();
        System.out.println("Insertar segundo #:");
        numero2 = valor.nextDouble();

        double resultado1 = numero1 + numero2;
        double resultado2 = numero1 - numero2;
        double resultado3 = numero1 * numero2;

        System.out.println("- la suma de "+numero1+" y "+numero2+" es: "+resultado1);
        System.out.println("- la resta de "+numero1+" y "+numero2+" es: "+resultado2);
        System.out.println("- la multiplicacion de "+numero1+" y "+numero2+" es: "+resultado3);

        division = numero1 / numero2;

        if(numero2 != 0){
            division = numero1 / numero2;
        }else{
            System.out.println("Error, digito no valido");
        }

        double resultado4 = division;

        System.out.println("- la division de "+numero1+" y "+numero2+" es: "+resultado4);
    }
}
```

1.2.

Escribir un programa que de la entrada estándar el precio de un producto y muestre en la salida estándar el precio del producto al aplicarle el IGV.

1.2.1. Código Pseint

Algoritmo CalcularPrecioconIGV

Definir precio,igv,precioConIGV Como Real

Escribir 'Ingrese el precio del producto:'

Leer precio

Escribir 'Ingrese el porcentaje de IGV:'

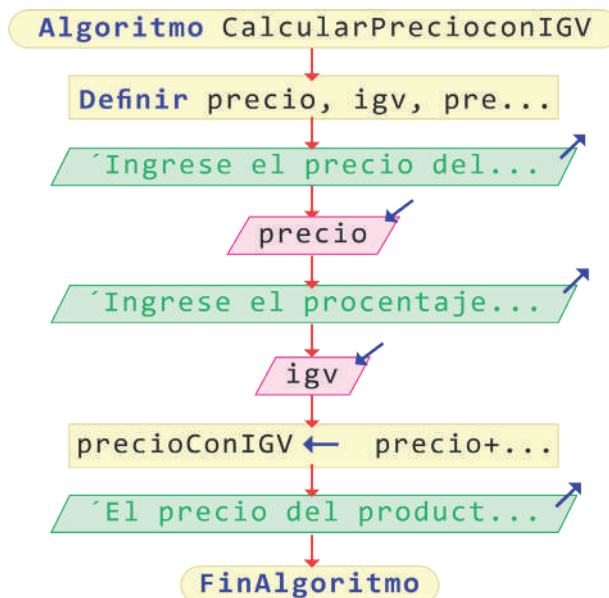
Leer igv

precioConIGV <- precio+(precio*igv/100)

Escribir 'El precio del producto con IGV es:',precioConIGV

FinAlgoritmo

1.2.2. Diagrama de flujo



1.2.3. Adaptación a Java

```
import java.util.Scanner;

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

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

        double igv = 0.18; // 18% de IGV
        double precioConIGV = precio * (1 + igv);

        System.out.println("El precio con IGV es: " + precioConIGV);
    }
}
```

1.3.

Realice un programa que lea de la entrada estándar los siguientes datos de una persona:

Edad: dato de tipo entero.

Sexo: dato de tipo carácter.

Altura en metros: dato de tipo real.

Tras leer los datos, el programa debe mostrarlos en la salida estándar.

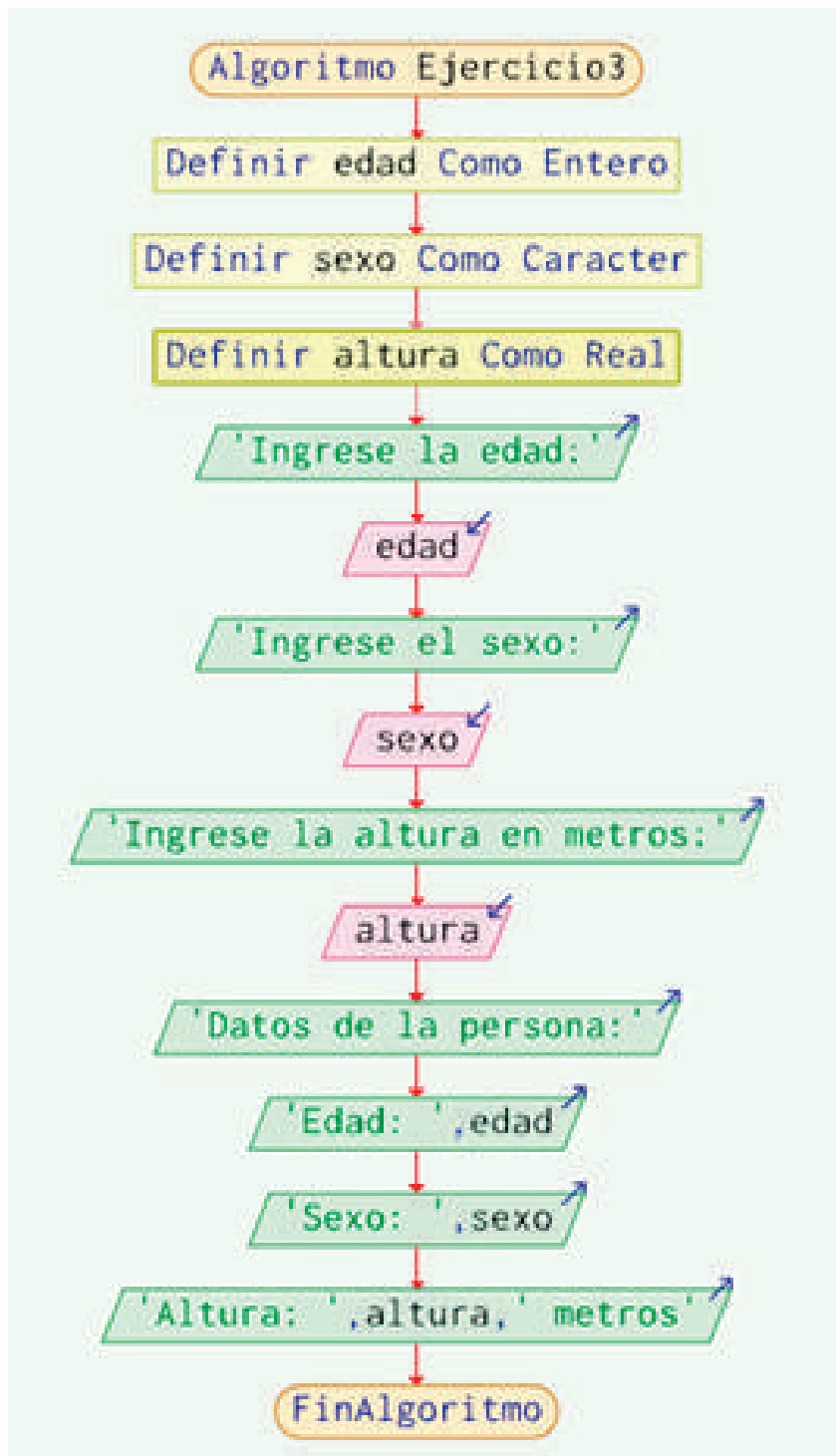
1.3.1. Código Pseint

Algoritmo Ejercicio3

Definir edad Como Entero
Definir sexo Como Caracter
Definir altura Como Real
Escribir "Ingrese la edad:"
Leer edad
Escribir "Ingrese el sexo:"
Leer sexo
Escribir "Ingrese la altura en metros:"
Leer altura
Escribir "Datos de la persona:"
Escribir "Edad:", edad
Escribir "Sexo:", sexo
Escribir "Altura:", altura, " metros"

FinAlgoritmo

1.3.2. Diagrama de flujo



1.3.3. Adaptación a Java

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        int edad;

        String genero;

        double altura;

        System.out.println("Ingrese su edad:");

        edad = entrada.nextInt();

        System.out.println("Ingrese su genero:");

        genero = entrada.next();

        System.out.println("Ingrese su altura:");

        altura = entrada.nextDouble();

        System.out.println("Su edad es "+edad+", tu tipo de genero es "+genero+" y su altura es "+altura);

    }

}
```


14.

Ejecute el programa del ejercicio anterior con entradas erróneas y observe los resultados. Por ejemplo, introduzca un dato de tipo carácter cuando se espera un dato de tipo entero.

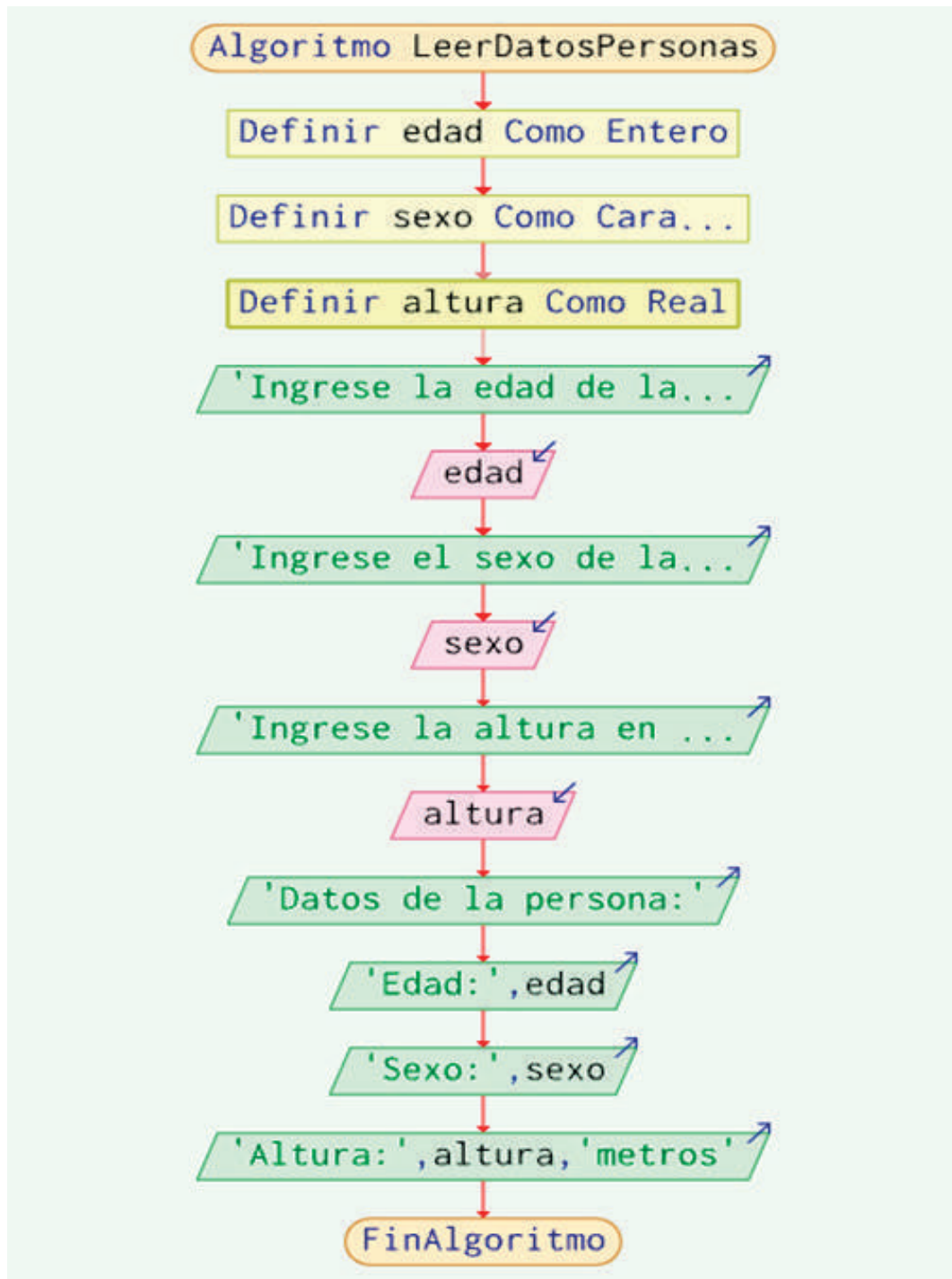
14.1. Código Pseint

Algoritmo LeerDatosPersonas

Definir edad Como Entero
Definir sexo Como Caracter
Definir altura Como Real
Escribir 'Ingrese la edad de la persona:'
Leer edad
Escribir 'Ingrese el sexo de la persona:'
Leer sexo
Escribir 'Ingrese la altura en metros:'
Leer altura
Escribir 'Datos de la persona:'
Escribir 'Edad:',edad
Escribir 'Sexo:',sexo
Escribir 'Altura:',altura,'metros'

FinAlgoritmo

14.2. Diagrama de flujo



14.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio4 {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        int edad = 0;
        try {
            System.out.print("Ingrese su edad:");
            edad = entrada.nextInt();
        } catch (Exception e) {
            System.out.println("Ha ingresado un valor no valido para la edad.");
        }
        char sexo = ' ';
        try {
            System.out.print("Ingrese su sexo (M/F):");
            sexo = entrada.next().charAt(0);
        } catch (Exception e) {
            System.out.println("Ha ingresado un valor no valido para el sexo.");
        }
        double altura = 0.0;
        try {
            System.out.print("Ingrese su altura en metros:");
            altura = entrada.nextDouble();
        } catch (Exception e) {
            System.out.println("Ha ingresado un valor no valido para la altura.");
        }
        System.out.println("Edad:" + edad);
        System.out.println("Sexo:" + sexo);
        System.out.println("Altura en metros:" + altura);
    }
}
```

Bloque 02: Expresiones y operadores

2.1.

Escribe la siguiente expresión en java y obtener el valor de la ecuación:

$$\text{a) } \frac{a}{b} + 1$$

2.1.1. Código Pseint

Algoritmo B2_Ejercicio_I

Definir a, b, resultado Como Real

Escribir “Escribir el valor de a y b y el programa mostrará la ecuación $a/b + 1$ “

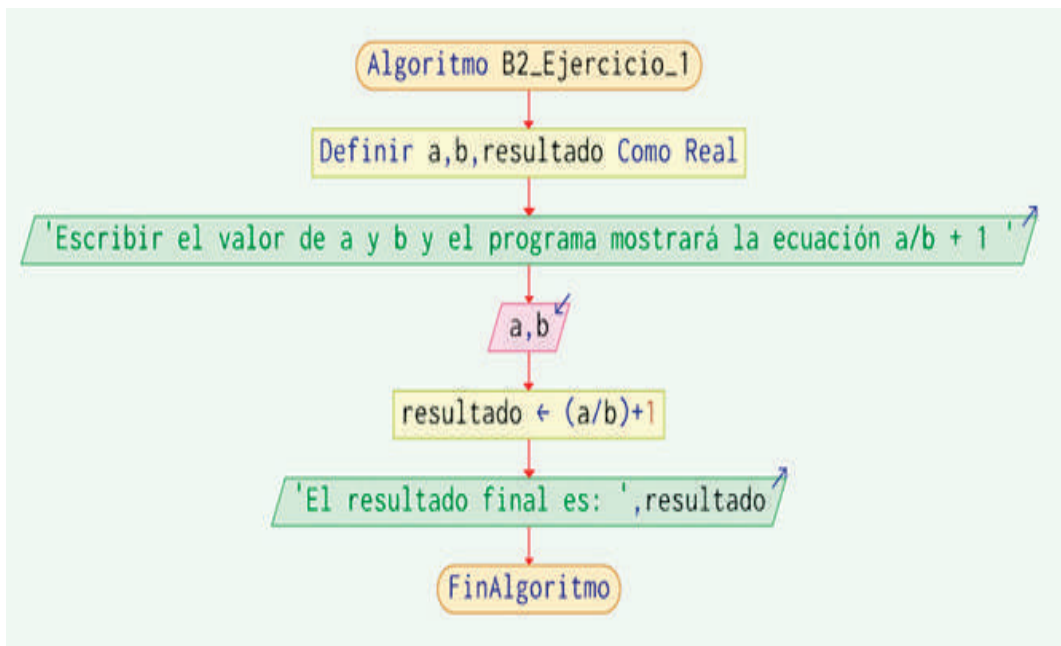
Leer a,b

resultado = $(a/b) + 1$

Escribir “El resultado final es:”, resultado

FinAlgoritmo

2.1.2. Diagrama de flujo



2.1.3. Adaptación a java

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) {
        int a,b, resultado;

        Scanner entrada = new Scanner(System.in);
        System.out.println("Digite el valor de a:");
        a = entrada.nextInt();

        System.out.println("Digite el valor de b:");
        b = entrada.nextInt();

        resultado = (a/b) + 1;

        System.out.println("El resultado es:" + resultado);
    }
}
```

2.2.

Escribe la siguiente expresión en java y obtener el valor de la ecuación:

$$\mathbf{b)} \frac{a+b}{c+d}$$

2.2.1. Código Pseint

Algoritmo Problema2

Definir a,b,c,d Como Entero

Definir r Como Real

Escribir “Progrma que realiza la suma de fracciones”;

Escribir “Digite un numero entero el valor del numerador de la primera fraccion”;

Leer a;

Escribir “Digite un numero entero el valor del numerador de la segunda fraccion”;

Leer b;

Escribir “Digite un numero entero el valor del numerador de la tercera fraccion”;

Leer c;

Escribir “Digite un numero entero el valor del numerador de la cuarta fraccion”;

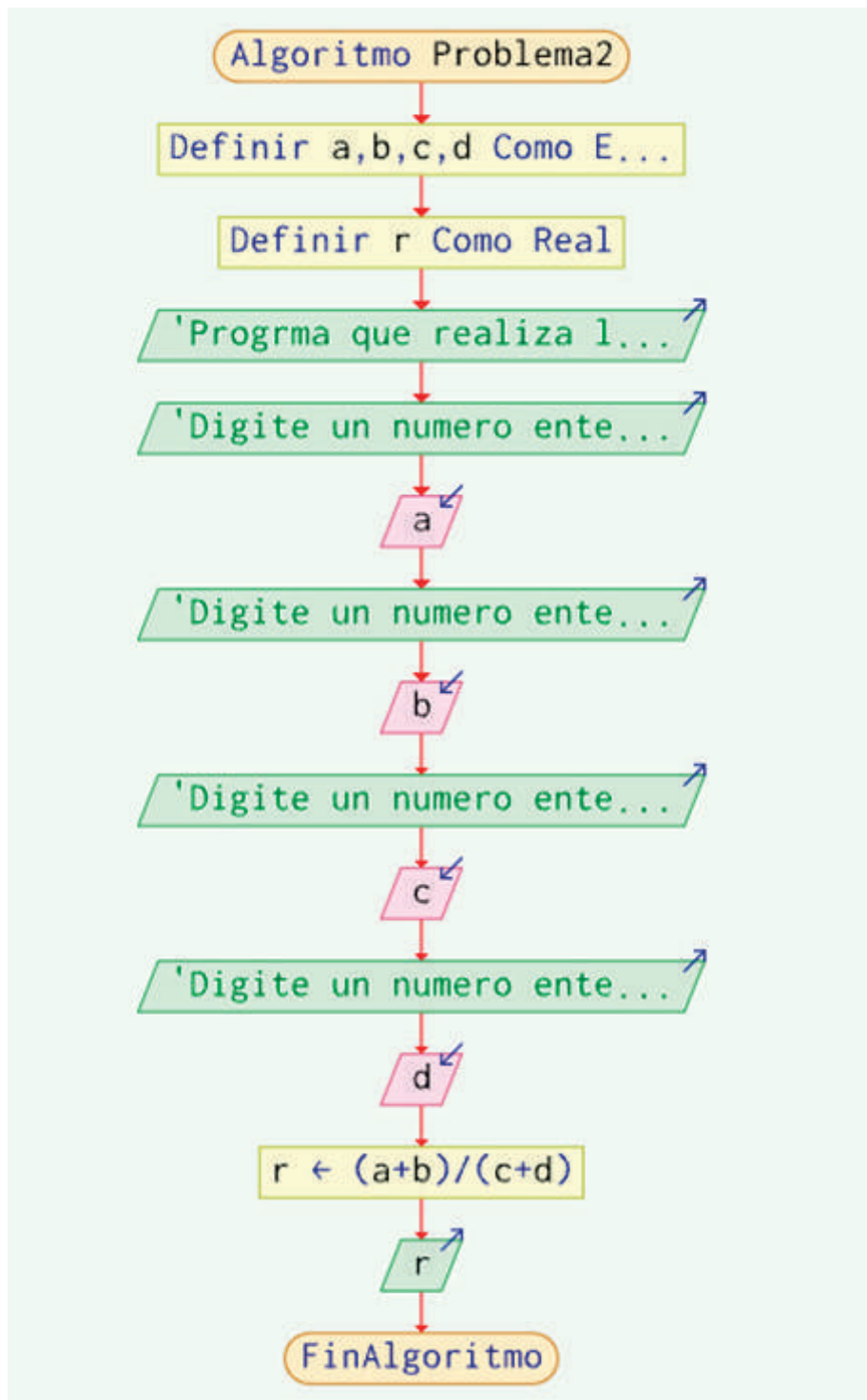
Leer d;

$r = (a+b)/(c+d);$

Escribir r;

FinAlgoritmo

2.2.2. Diagrama de flujo



2.3.3 Adaptación a java

```
import java.util.Scanner;
public class CalculadoraExpresion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el valor de a:");
        double a = scanner.nextDouble();
        System.out.print("Ingrese el valor de b:");
        double b = scanner.nextDouble();
        System.out.print("Ingrese el valor de c:");
        double c = scanner.nextDouble();
        System.out.print("Ingrese el valor de d:");
        double d = scanner.nextDouble();
        double resultado = (a + b) / (c + d);
        System.out.println("El resultado de la expresión es: " + resultado);
    }
}
```


2.3.

Escribe la siguiente expresión en java y obtener el valor de la ecuación:

$$c) \frac{a + \frac{b}{c}}{d + \frac{e}{f}}$$

2.3.1. Código Pseint

Algoritmo B2_Ejercicio_3

Definir a,b,c,d,e,f, resultado Como Real

Escribir "Escribir el valor de a,b,c,d,e,f y el programa mostrará la ecuación (a+b/c)/(d+e/f) "

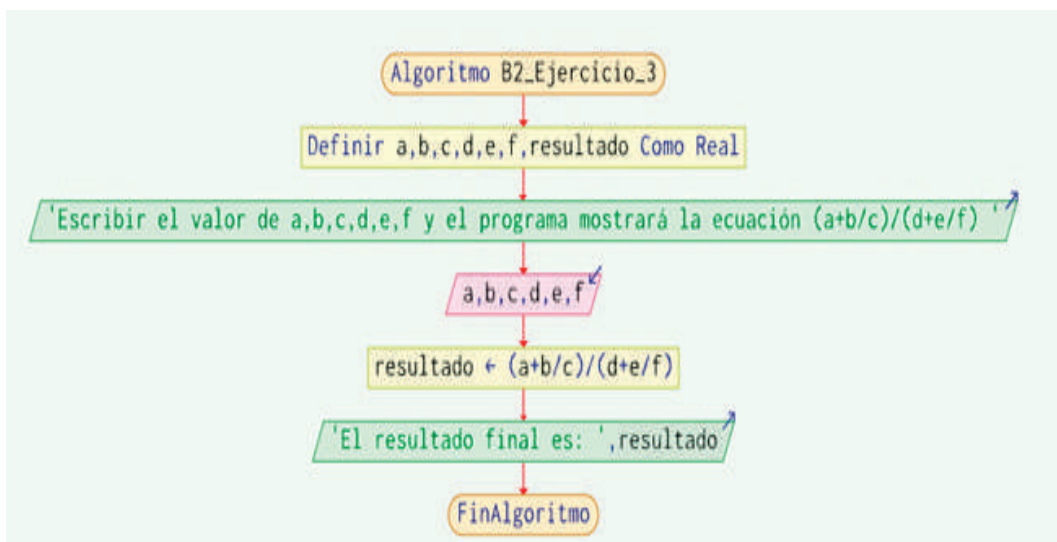
Leer a,b,c,d,e,f

resultado = (a+b/c)/(d+e/f)

Escribir "El resultado final es:", resultado

FinAlgoritmo

2.3.2. Diagrama de flujo



2.3.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio3bloque2 {
    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        double a,b,c,d,e,f,resultado;
        System.out.println("Ingrese el valor de a:");
        a = sc.nextDouble();
        System.out.println("Ingrese el valor de b:");
        b = sc.nextDouble();
        System.out.println("Ingrese el valor de c ");
        c = sc.nextDouble();
        System.out.println("Ingrese el valor de d");
        d = sc.nextDouble();
        System.out.println("Ingrese el valor de e");
        e = sc.nextDouble();
        System.out.println("Ingrese el valor de f");
        f = sc.nextDouble();
        resultado = (a + (b / c)) / (d + (e / f));
        System.out.println("El resultado de la expresión es:" + resultado);
    }
}
```

24.

Escribe la siguiente expresión como expresión en java:

$$\text{d) } a + \frac{b}{c-d}$$

24.1. Código Pseint

Algoritmo Problema2

Definir a,b,c,d Como Entero

Definir r Como Real

Escribir "Programa que realiza la suma de fracciones";

Escribir "Digite un numero entero el valor del numerador de la primera fraccion";

Leer a;

Escribir "Digite un numero entero el valor del numerador de la segunda fraccion";

Leer b;

Escribir "Digite un numero entero el valor del numerador de la tercera fraccion";

Leer c;

Escribir "Digite un numero entero el valor del numerador de la cuarta fraccion";

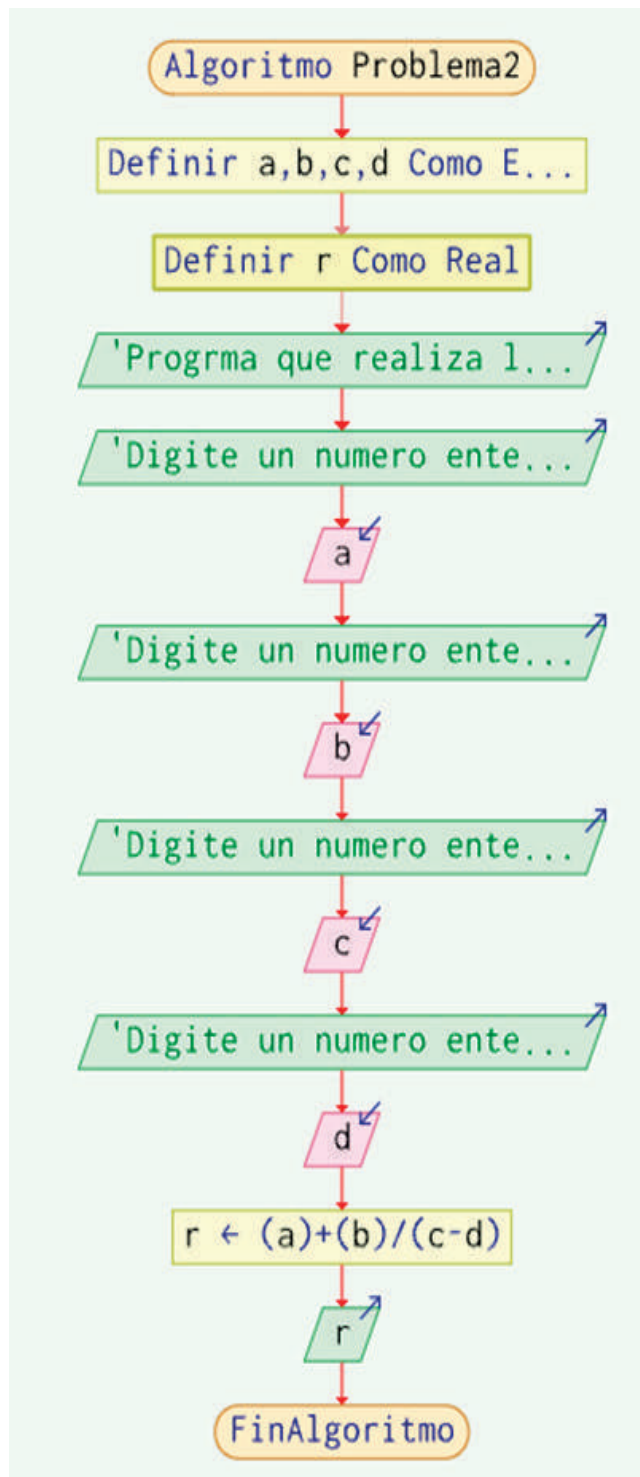
Leer d;

$r = (a) + (b)/(c-d);$

Escribir r;

FinAlgoritmo

24.2. Diagrama de flujo



24.3. Adaptación a Java

```
import java.util.Scanner;
public class CalculadoraExpresion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el valor de a:");
        double a = scanner.nextDouble();
        System.out.print("Ingrese el valor de b:");
        double b = scanner.nextDouble();
        System.out.print("Ingrese el valor de c:");
        double c = scanner.nextDouble();
        System.out.print("Ingrese el valor de d:");
        double d = scanner.nextDouble();
        double resultado = a + (b / c - d);
        System.out.println("El resultado de la expresión es:" + resultado);
    }
}
```

2.5.

Escriba un fragmento de programa que intercambie los valores de dos variables.

2.5.1. Código Pseint

Algoritmo B2_Ejercicio_5

Definir a, b, aux como Real

Escribir "Ingresar valor de a"

Leer a

Escribir "Ingresar valor de a"

Leer b

Escribir "El valor inicial de a es:", a

Escribir "El valor inicial de b es:", b

aux=a

a=b

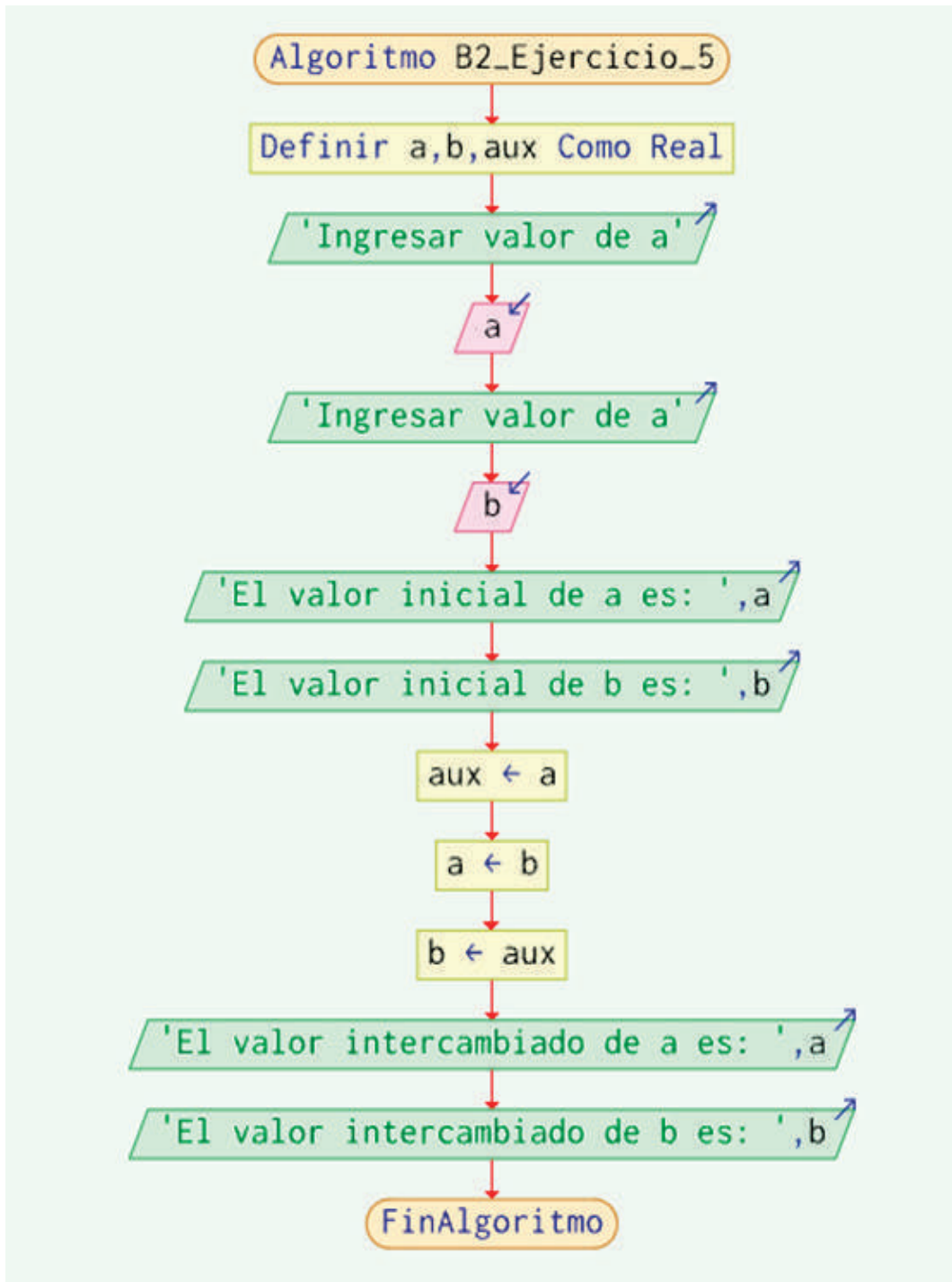
b=aux

Escribir "El valor intercambiado de a es:", a

Escribir "El valor intercambiado de b es:", b

FinAlgoritmo

2.5.2. Diagrama de flujo



2.5.3. Adaptación a Java

```
public class Ejercicio5bloque2 {  
  
    public static void main(String[] args) {  
        int a = 8;  
        int b = 16;  
        System.out.println("Antes del intercambio:");  
        System.out.println("a = " + a + ", b = " + b);  
        int temp = a;  
  
        a = b;  
        b = temp;  
        System.out.println("Después del intercambio");  
        System.out.println("a = " + a + ", b = " + b);  
  
    }  
}
```

2.6.

Escriba un programa que lea las tres notas de un alumno y calcule la nota final media de dicho alumno.

2.6.1. Código Pseint

Algoritmo CalcularNotaFinalMedia

Definir nota1, nota2, nota3, notaFinalMedia Como Real

Escribir "Ingrese la nota 1:"

Leer nota1

Escribir "Ingrese la nota 2:"

Leer nota2

Escribir "Ingrese la nota 3:"

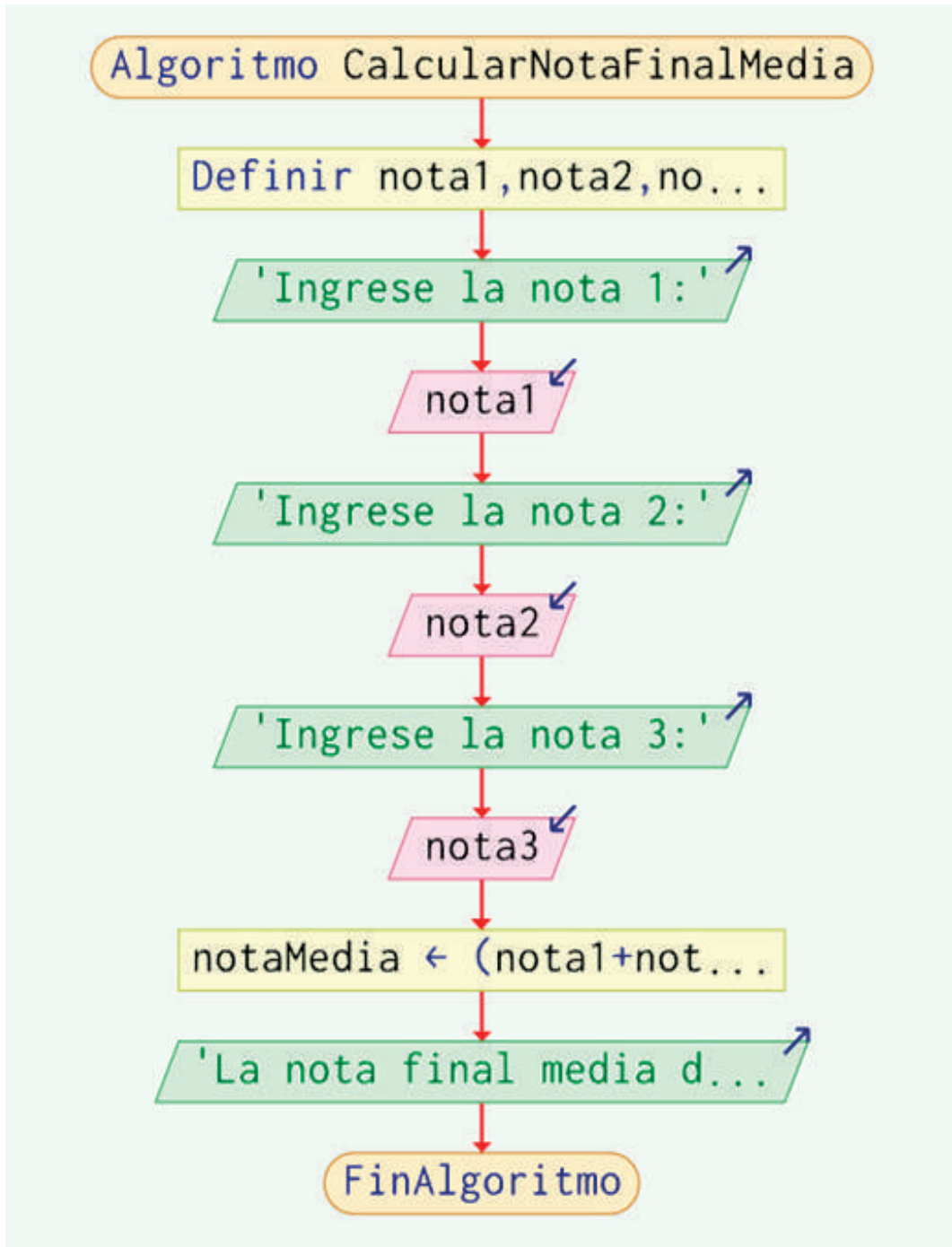
Leer nota3

notaMedia <- (nota1 + nota2 + nota3) / 3

Escribir "La nota final media del alumno es:", notaMedia

FinAlgoritmo

2.6.2. Diagrama de flujo



2.6.3. Adaptación a Java

```
import java.util.Scanner;

public class CalculadoraNotaMedia {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese la primera nota:");

        double nota1 = scanner.nextDouble();

        System.out.print("Ingrese la segunda nota:");

        double nota2 = scanner.nextDouble();

        System.out.print("Ingrese la tercera nota:");

        double nota3 = scanner.nextDouble();

        double notaMedia = (nota1 + nota2 + nota3) / 3;

        System.out.println("La nota final media del alumno es:" + notaMedia);

    }

}
```

2.7.

La calificación final de un estudiante es el promedio de tres notas: la nota de prácticas que cuenta un 30% del total, la nota teórica que cuenta un 60% y la nota de participación que cuenta el 10% restante. Escriba un programa que lea las tres notas del alumno y escriba su nota final.

2.7.1. Código Pseint

Algoritmo B2_Ejercicio_7

Definir Nota1, Nota2, Nota3, Promedio Como Real

Escribir "Ingresar nota de práctica"

Leer Nota1

Escribir "Ingresar nota teórica"

Leer Nota2

Escribir "Ingresar nota de participación"

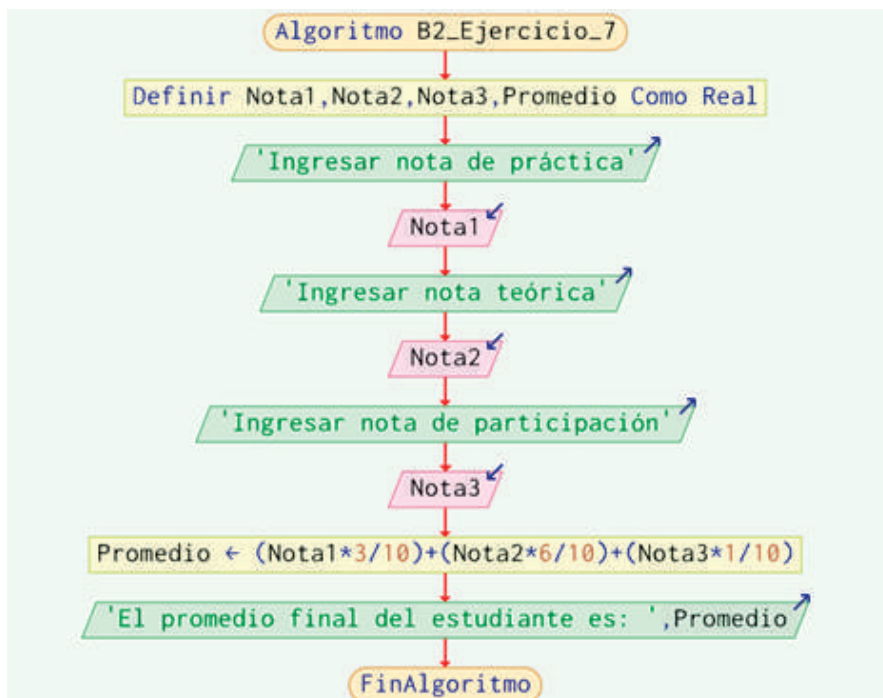
Leer Nota3

Promedio = $(Nota1 * 3/10) + (Nota2 * 6/10) + (Nota3 * 1/10)$

Escribir "El promedio final del estudiante es:", Promedio

FinAlgoritmo

2.7.2. Diagrama de flujo



2.7.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio7 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double notaUno,notaDos,notaTres,promedio;
        String Estudiante ;
        final int K=30;
        final int B=60;
        final int C=10;
        System.out.println("Nombre del Estudiante:");
        Estudiante= teclado.nextLine();
        System.out.println("Ingrese la nota de Practica:");
        notaUno= teclado.nextDouble();
        System.out.println("Ingrese la nota de Teorica:");
        notaDos= teclado.nextDouble();
        System.out.println("Ingrese la nota de Participacion:");
        notaTres= teclado.nextDouble();
        promedio= (((notaUno*K)+(notaDos*B)+(notaTres*C))/100);
        System.out.println("La nota final del estudiante "+Estudiante+" es de "+promedio);
    }
}
```

2.8.

Escriba un programa que lea de la entrada estándar los dos catetos de un triángulo rectángulo y escriba en la salida estándar su hipotenusa.

2.8.1. Código Pseint

Algoritmo CalcularTriánguloRectángulo

Definir cateto1, cateto2, hipotenusa Como Real

Escribir "Ingrese el valor del primer cateto:"

Leer cateto1

Escribir "Ingrese el valor del segundo cateto:"

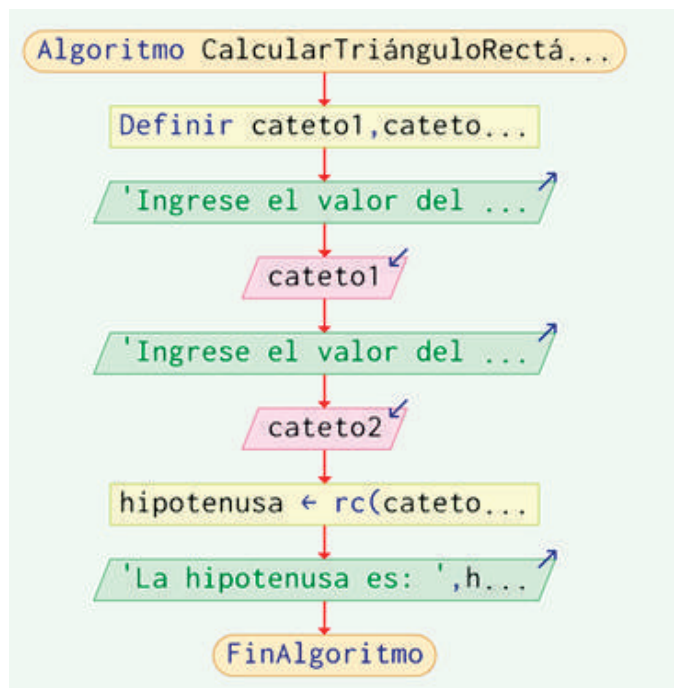
Leer cateto2

hipotenusa \leftarrow rc(cateto1² + cateto2²)

Escribir "La hipotenusa es:", hipotenusa

FinAlgoritmo

2.8.2. Diagrama de flujo



2.8.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio8Bloque2{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese la longitud del primer cateto:");

        double cateto1 = scanner.nextDouble();

        System.out.print("Ingrese la longitud del segundo cateto:");

        double cateto2 = scanner.nextDouble();

        double hipotenusa = Math.sqrt(Math.pow(cateto1, 2) + Math.pow(cateto2, 2));

        System.out.println("La hipotenusa del triángulo rectángulo es:" + hipotenusa);

    }

}
```

2.9.

Realice un programa que calcule el valor que toma la siguiente función para unos valores dados de x e y:

$$f(x, y) = \frac{\sqrt{x}}{y^2 - 1}$$

2.9.1. Código Pseint

Algoritmo B2_Ejercicio_9

Definir x,z, resultado Como Real

Escribir "Ingresar el valor de (x, y). El programa mostrará la función $f(x,y) = \text{raiz}(x)/y^2 - 1$ "

Escribir "Ingresar el valor de x:"

Leer x

Escribir "Ingresar el valor de y:"

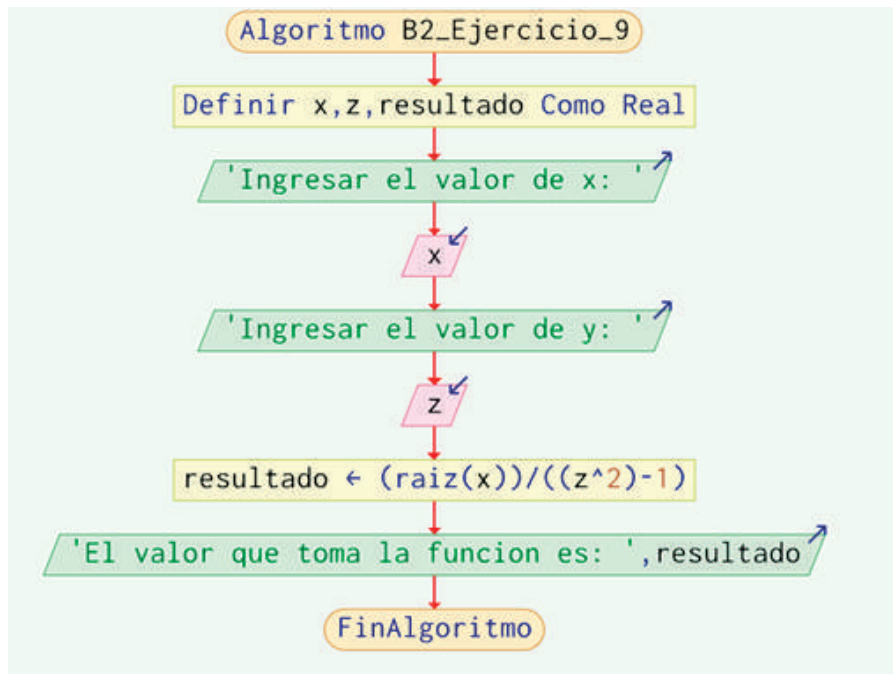
Leer z

resultado = (raiz(x))/((z^2)-1)

Escribir "El valor que toma la funcion es:", resultado

FinAlgoritmo

2.9.2. Diagrama de flujo



2.9.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio9Bloque2 {

    public static void main(String[] args) {
        float ejeX;
        float ejeY;
        float divisor;
        Scanner entrada = new Scanner(System.in);

        System.out.println("La función es  $f(x,y) = [(x^{1/2})]/[(y^2)-1]$ ");
        System.out.println("Digite el valor para x:");
        ejeX = entrada.nextFloat();

        while (ejeX < 0){

            System.out.println("ERROR, Digite el valor para x nuevamente:");

            ejeX = entrada.nextFloat();

        }

        double raiz = Math.pow(ejeX, 0.5);

        System.out.println("Ahora digite el valor para y:");

        ejeY = entrada.nextFloat();

        divisor = ejeY*ejeY - 1;

        while (divisor == 0){
            System.out.println("ERROR, digite nuevamente el valor para y:");
            ejeY = entrada.nextFloat();
            divisor = ejeY*ejeY - 1;
        }

        System.out.println("El valor que toma la función es: " + (raiz / divisor));

    }

}
```


2.10.

Escriba un programa que calcule las soluciones de una ecuación de segundo grado de la forma $ax^2 + bx + c = 0$, teniendo en cuenta que:

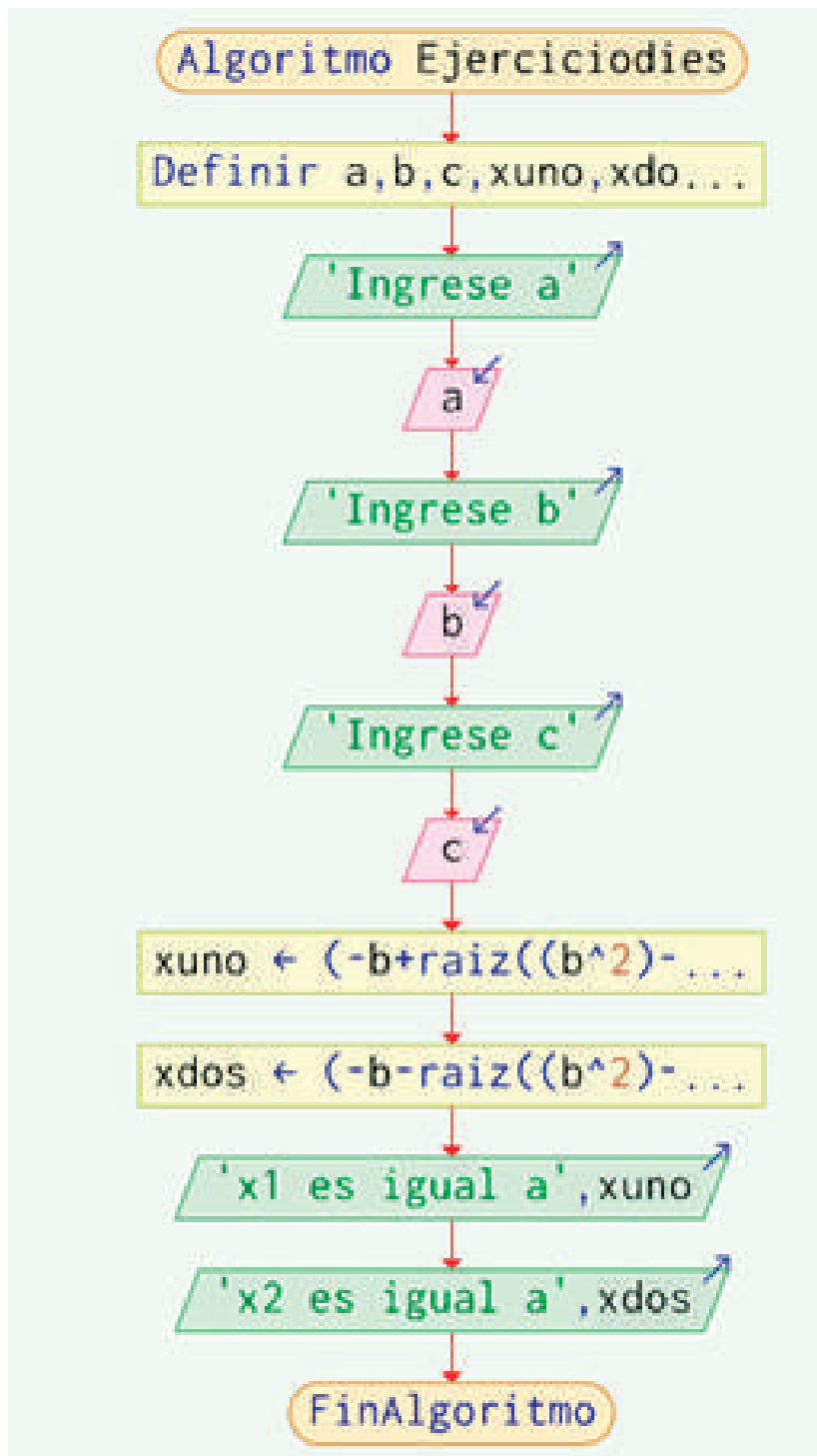
2.10.1. Código Pseint

Algoritmo Ejerciciodies

```
Definir a,b,c, xuno, xdos Como Real;  
Escribir "Ingrese a";  
leer a;  
Escribir "Ingrese b";  
leer b;  
Escribir "Ingrese c";  
leer c;  
xuno = (- b + raiz( (b^2) - (4*a*c))) / 2*a;  
xdos = (- b - raiz( (b^2) - (4*a*c))) / 2*a;  
Escribir "x1 es igual a", xuno;  
Escribir "x2 es igual a", xdos;
```

FinAlgoritmo

2.10.2. Diagrama de flujo



2.10.3. Adaptación a Java

```
import java.util.Scanner;

public class CalculadoraEcuacionSegundoGrado {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el coeficiente a:");
        double a = scanner.nextDouble();
        System.out.print("Ingrese el coeficiente b:");
        double b = scanner.nextDouble();
        System.out.print("Ingrese el coeficiente c:");
        double c = scanner.nextDouble();
        double discriminante = Math.pow(b, 2) - (4 * a * c);
        if (discriminante > 0) {
            double x1 = (-b + Math.sqrt(discriminante)) / (2 * a);
            double x2 = (-b - Math.sqrt(discriminante)) / (2 * a);
            System.out.println("Las soluciones son x1 = " + x1 + " y x2 = " + x2);
        } else if (discriminante == 0) {
            double x = -b / (2 * a);
            System.out.println("La solución es x = " + x);
        } else {
            System.out.println("La ecuación no tiene soluciones reales.");
        }
    }
}
```

Bloque 03: Estructuras condicionales

3.1.

Escriba un programa que lea dos números y determine cuál de ellos es el mayor.

3.1.1. Código Pseint

Algoritmo B3_Ejercicio_1

Definir nro1, nro2, max Como Real

Escribir “Ingresar dos números y el programa mostrará el mayor de ambos”

Escribir “Ingresar primer número”

Leer nro1

Escribir “Ingresar segundo número”

Leer nro2

Si $nro1 > nro2$ Entonces

Escribir “El número mayor es:“, nro1

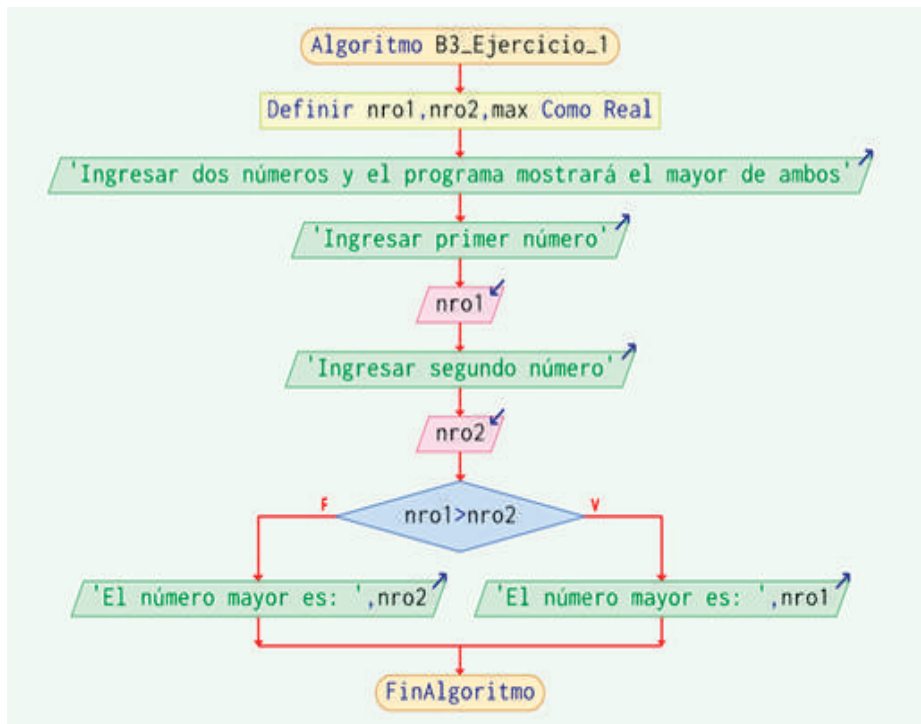
SiNo

Escribir “El número mayor es:“, nro2

Fin Si

FinAlgoritmo

3.1.2. Diagrama de flujo



3.1.3. Adaptación a Java

```

import java.util.Scanner;

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

        System.out.println("Digite el primer número");
        double primero = entrada.nextDouble();
        System.out.println("Digite el segundo número");

        double segundo = entrada.nextDouble();
        if(primero > segundo){
            System.out.println("el mayor es:" + primero);        } else{
            System.out.println("El mayor es:" + segundo);
        }
    }
}
  
```

3.2.

Escriba un programa que lea tres números y determine cuál de ellos es el mayor.

3.2.1. Código Pseint

Algoritmo NumeroMayor

Definir numero1, numero2, numero3, mayor Como Entero

Escribir "Ingrese el primer número:"

Leer numero1

Escribir "Ingrese el segundo número:"

Leer numero2

Escribir "Ingrese el tercer número:"

Leer numero3

mayor <- numero1

Si numero2 > mayor Entonces

 mayor <- numero2

FinSi

Si numero3 > mayor Entonces

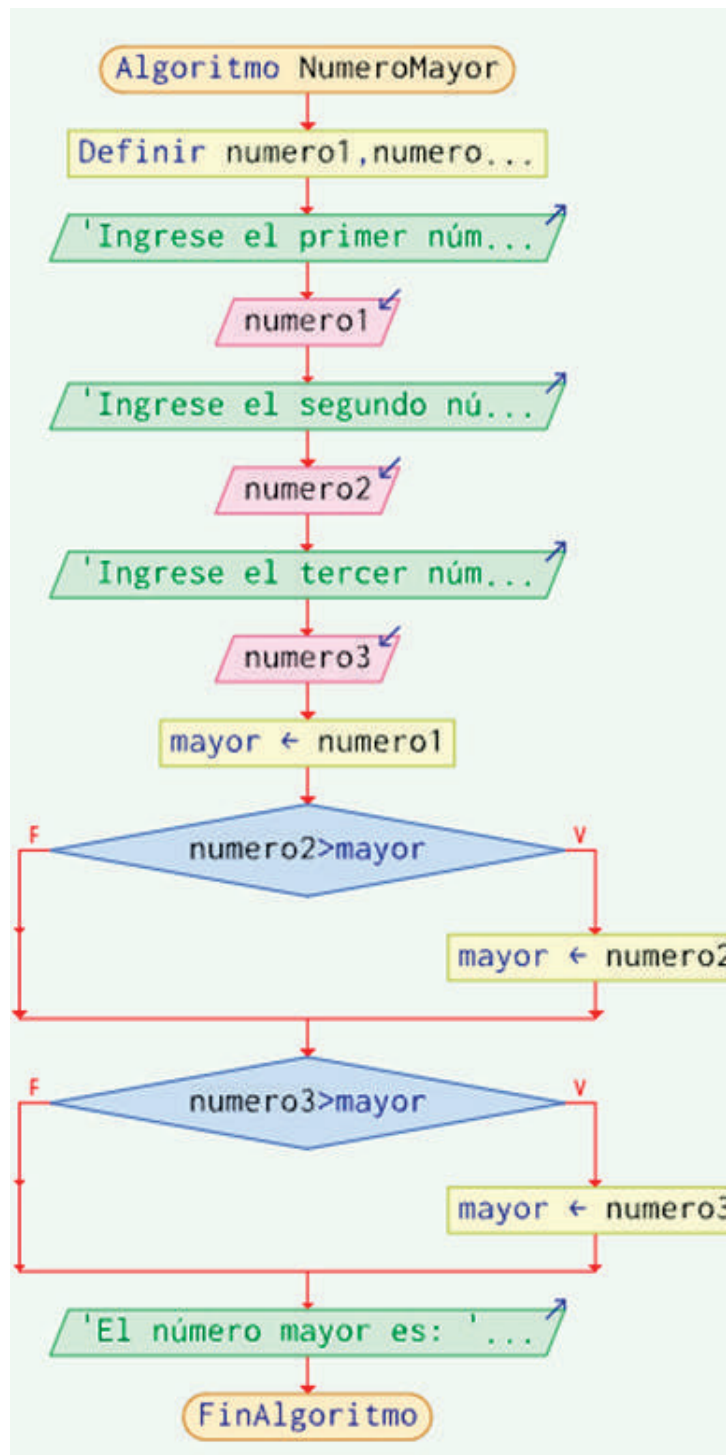
 mayor <- numero3

FinSi

Escribir "El número mayor es:", mayor

FinAlgoritmo

3.2.2. Diagrama de flujo



3.2.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio2Bloque3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el primer número:");

        int numero1 = scanner.nextInt();

        System.out.print("Ingrese el segundo número:");

        int numero2 = scanner.nextInt();

        System.out.print("Ingrese el tercer número:");

        int numero3 = scanner.nextInt();

        int mayor = numero1; // Asignamos el primer número como el mayor inicialmente

        if (numero2 > mayor) {

            mayor = numero2;

        }

        if (numero3 > mayor) {

            mayor = numero3;

        }

        // Mostrar el número mayor

        System.out.println("El número mayor es: " + mayor);

    }

}
```


3.3.

Realice un programa que lea un valor entero y determine si se trata de un número par o impar.

3.3.1. Código Pseint

Algoritmo B3_Ejercicio_3

Definir nro Como Real

Escribir "Ingresar un valor entero y el programa determinará si se trata de un número par o impar."

Leer nro

Si $nro \text{ MOD } 2 == 0$ Entonces

Escribir "El número es par"

SiNo

Escribir "El número es impar"

Fin Si

FinAlgoritmo

3.3.2. Diagrama de flujo



3.3.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio3Bloque3 {

    public static void main(String[] args) {

        int numero, mitadNumero;

        Scanner entrada = new Scanner(System.in);

        System.out.println("Digite el número:");

        numero = entrada.nextInt();

        mitadNumero = numero % 2;

        if (mitadNumero == 0){

            System.out.println("El número es par");

        } else{

            System.out.println("El número es impar");

        }

    }

}
```

34.

Comprobar si un número digitado por el usuario es positivo o negativo

34.1. Código Pseint

Algoritmo Verificación

Definir numero Como Real

Escribir "Ingrese un número:"

Leer numero

Si numero > 0 Entonces

 Escribir "El número es positivo"

FinSi

Si numero < 0 Entonces

 Escribir "El número es negativo"

FinSi

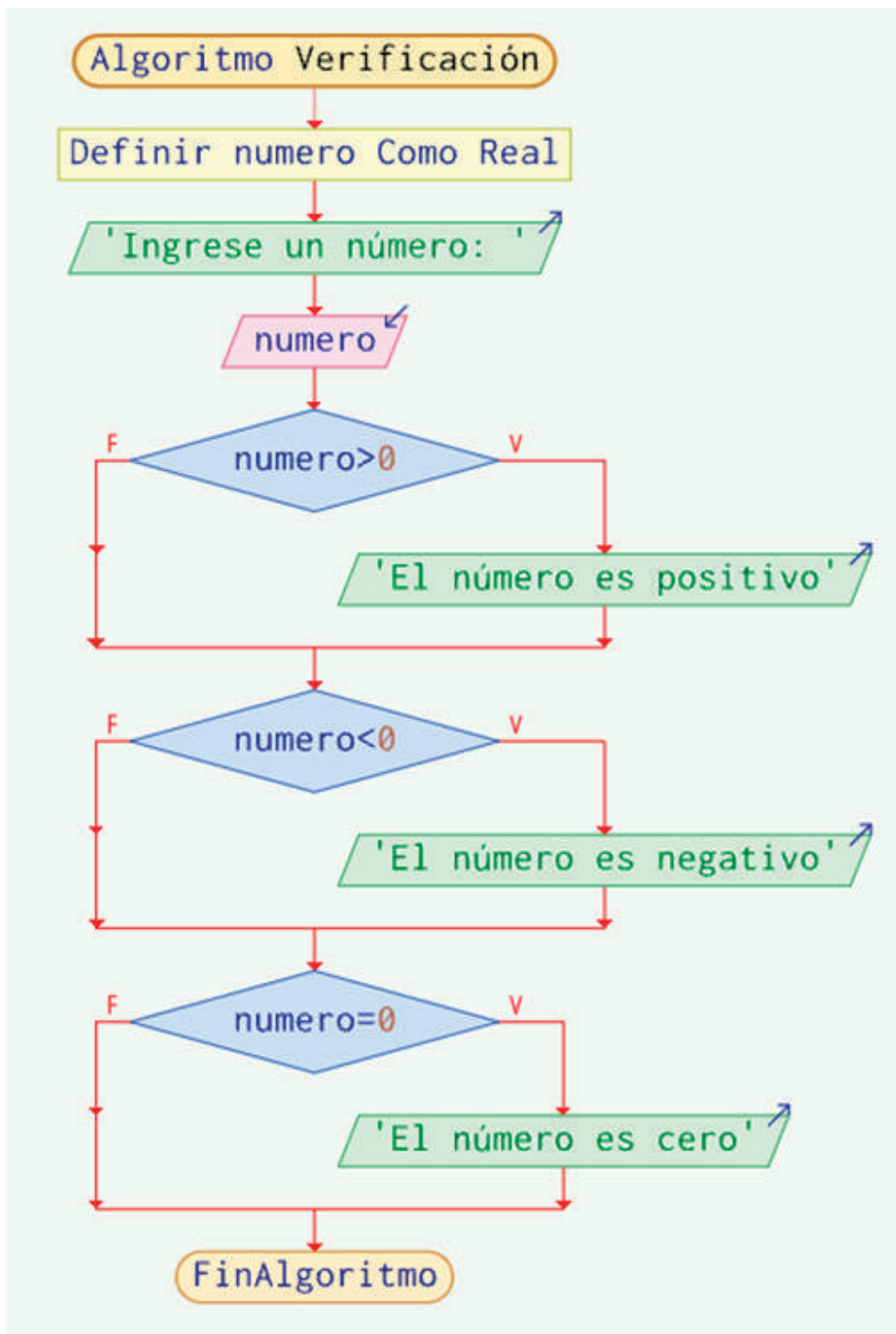
Si numero = 0 Entonces

 Escribir "El número es cero"

FinSi

FinAlgoritmo

34.2. Diagrama de flujo



34.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio4Bloque3{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese un número:");
        double numero = scanner.nextDouble();
        if (numero > 0) {
            System.out.println("El número ingresado es positivo.");
        } else if (numero < 0) {
            System.out.println("El número ingresado es negativo.");
        } else {
            System.out.println("El número ingresado es cero.");
        }
    }
}
```

3.5.

Escriba un programa que lea de la entrada estándar un carácter e indique en la salida estándar si el carácter es una vocal minúscula o no.

3.5.1. Código Pseint

Algoritmo B3_Ejercicio_5

Definir caracter Como Caracter

Escribir "Ingrese un carácter:"

Leer caracter

Si caracter == "a" O caracter == "e" O caracter == "i" O caracter == "o" O caracter == "u" Entonces

Escribir "El carácter ingresado es una vocal minúscula."

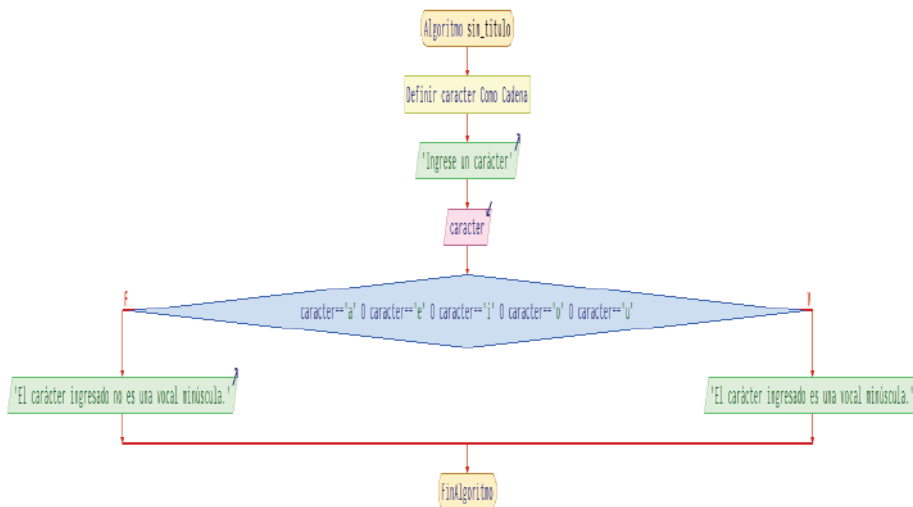
Sino

Escribir "El carácter ingresado no es una vocal minúscula."

FinSi

FinAlgoritmo

3.5.2. Diagrama de flujo



3.5.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio5Bloque3 {
    public static void main(String[] args) {
        char character;

        Scanner entrada = new Scanner(System.in);
        System.out.println("Digite un caracter:");
        character = entrada.next().charAt(0);
        if (Character.isUpperCase(character)){
            System.out.println(character + " es mayúscula");
        } else{
            System.out.println(character + " es minúscula");
        }
    }
}
```

3.6.

Escriba un programa que lea de la entrada estándar un carácter e indique en la salida estándar si el carácter es una vocal minúscula, es una vocal mayúscula o no es una vocal.

3.6.1. Código Pseint

Algoritmo CaracterVocal

Definir caracter Como Caracter

Escribir "Ingrese un carácter:"

Leer caracter

Si caracter = "a" o caracter = "e" o caracter = "i" o caracter = "o" o caracter = "u" Entonces

Escribir "El carácter es una vocal minúscula"

FinSi

Si caracter = "A" o caracter = "E" o caracter = "I" o caracter = "O" o caracter = "U" Entonces

Escribir "El carácter es una vocal mayúscula"

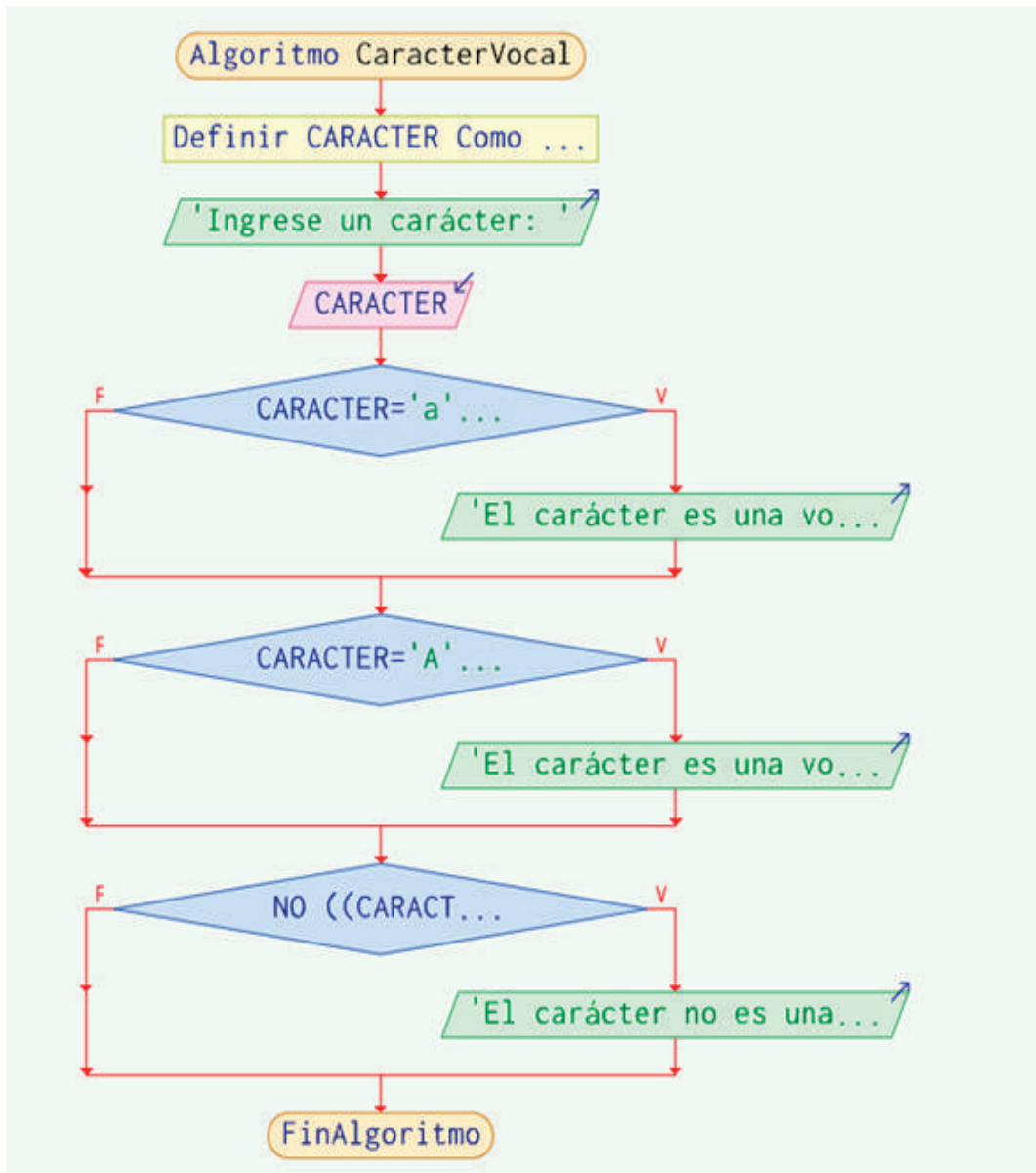
FinSi

Si No ((caracter >= "a" y caracter <= "z") o (caracter >= "A" y caracter <= "Z")) Entonces

Escribir "El carácter no es una vocal"

FinSi

3.6.2. Diagrama de flujo



3.6.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio6bloque3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese un carácter:");

        char character = scanner.nextLine().charAt(0);

        if (esVocalMinuscula(character)) {

            System.out.println("El carácter ingresado es una vocal minúscula.");

        } else if (esVocalMayuscula(character)) {

            System.out.println("El carácter ingresado es una vocal mayúscula.");

        } else {

            System.out.println("El carácter ingresado no es una vocal.");

        }

    }

    public static boolean esVocalMinuscula(char c) {

        return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';

    }

    public static boolean esVocalMayuscula(char c) {

        return c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U';

    }

}
```

3.7.

Escriba un programa que solicite una edad (un entero) e indique en la salida estándar si la edad introducida está en el rango [18-25].

3.7.1. Código Pseint

Algoritmo B3_Ejercicio_7

Definir edad Como Entero

Escribir "El programa indicará si la edad ingresada está en el rango de 18-25"

Escribir "Ingresar edad"

Leer edad

Si edad \geq 18 y edad \leq 25 Entonces

Escribir "La edad ", edad, " está dentro del rango 18-25"

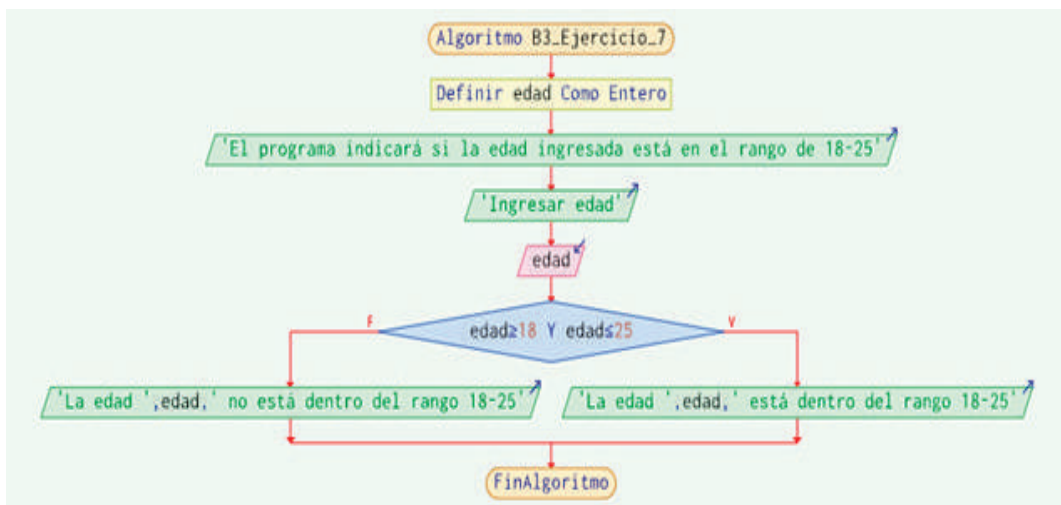
SiNo

Escribir "La edad ", edad, " no está dentro del rango 18-25"

Fin Si

FinAlgoritmo

3.7.2. Diagrama de flujo



3.7.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio7Bloque3 {
    public static void main(String[] args) {
        int edad;

        Scanner entrada = new Scanner(System.in);

        System.out.println("Digite su edad:");

        edad = entrada.nextInt();

        if (edad >= 18 && edad <= 25){
            System.out.println("Está dentro del rango, bienvenido");
        } else{
            System.out.println("No está dentro del rango");
        }
    }
}
```

3.8.

Escribe un programa que lea de la entrada estándar tres números. Después debe leer un cuarto número e indicar si el número coincide con alguno de los introducidos con anterioridad.

3.8.1. Código Pseint

Algoritmo Inspeccionar

Definir num1, num2, num3, num4 Como Entero

Escribir "Ingrese el primer número:"

Leer num1

Escribir "Ingrese el segundo número:"

Leer num2

Escribir "Ingrese el tercer número:"

Leer num3

Escribir "Ingrese el cuarto número:"

Leer num4

Si num4 = num1 o num4 = num2 o num4 = num3 Entonces

Escribir "El número ", num4, " coincide con uno de los números anteriores."

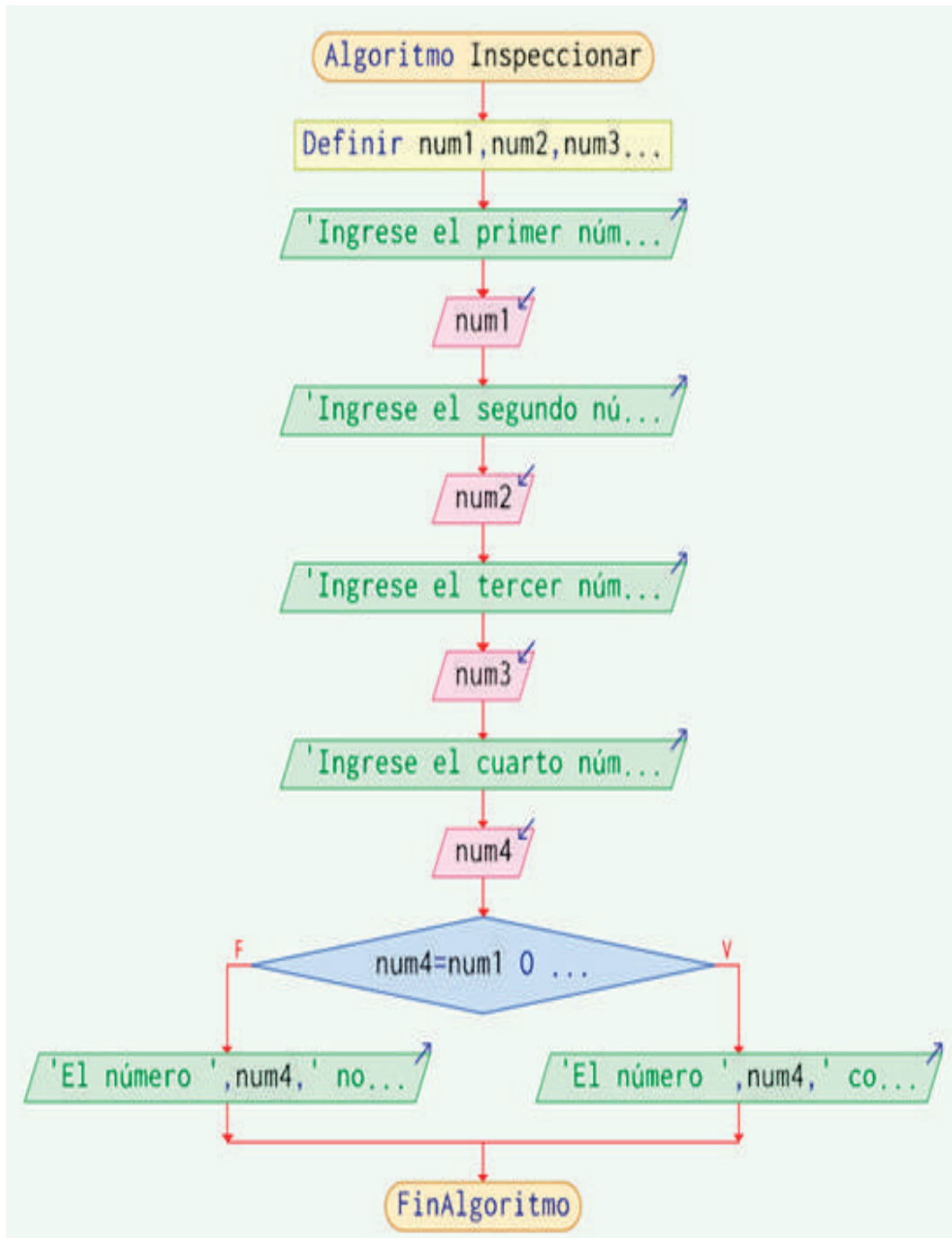
Sino

Escribir "El número ", num4, " no coincide con ninguno de los números anteriores."

FinSi

FinAlgoritmo

3.8.2. Diagrama de flujo



3.8.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio8Bloque3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el primer número:");
        int numero1 = scanner.nextInt();
        System.out.print("Ingrese el segundo número:");
        int numero2 = scanner.nextInt();
        System.out.print("Ingrese el tercer número:");
        int numero3 = scanner.nextInt();
        System.out.print("Ingrese el cuarto número:");
        int numero4 = scanner.nextInt();
        if (numero4 == numero1 || numero4 == numero2 || numero4 == numero3) {
            System.out.println("El cuarto número coincide con uno de los números
            introducidos anteriormente.");
        } else {
            System.out.println("El cuarto número no coincide con ninguno de los núme-
            ros introducidos anteriormente.");
        }
    }
}
```

3.9.

Cambiar un número entero con el mismo valor pero en romanos.

3.9.1. Código Pseint

Algoritmo B3_Ejercicio_9

Definir a,c,d,u,n Como Entero

Escribir “Ingresar un número del 1-3999 y el programa lo convertirá a romanos”

Escribir “Escribe un numero”

leer n

$m = \text{trunc}(n/1000) \bmod 10$

$c = \text{trunc}(n/100) \bmod 10$

$d = \text{trunc}(n/10) \bmod 10$

$u = \text{trunc}(n/1) \bmod 10$

si $m > 3$ Entonces

Escribir “El numero no se puede calcular”

SiNo

segun m hacer

- 1: Escribir “M” Sin Saltar
- 2: Escribir “MM” Sin Saltar
- 3: Escribir “MMM” Sin Saltar

FinSegun

segun c hacer

- 1: Escribir "C" Sin Saltar
- 2: Escribir "CC" Sin Saltar
- 3: Escribir "CCC" Sin Saltar
- 4: Escribir "CD" Sin Saltar
- 5: Escribir "D" Sin Saltar
- 6: Escribir "DC" Sin Saltar
- 7: Escribir "DCC" Sin Saltar
- 8: Escribir "DCCC" Sin Saltar
- 9: Escribir "CM" Sin Saltar

FinSegun

Segun d hacer

- 1: Escribir "X" Sin Saltar
- 2: Escribir "XX" Sin Saltar
- 3: Escribir "XXX" Sin Saltar
- 4: Escribir "XL" Sin Saltar
- 5: Escribir "L" Sin Saltar
- 6: Escribir "LX" Sin Saltar
- 7: Escribir "LXX" Sin Saltar
- 8: Escribir "LXXX" Sin Saltar
- 9: Escribir "XC" Sin Saltar

FinSegun

Segun u hacer

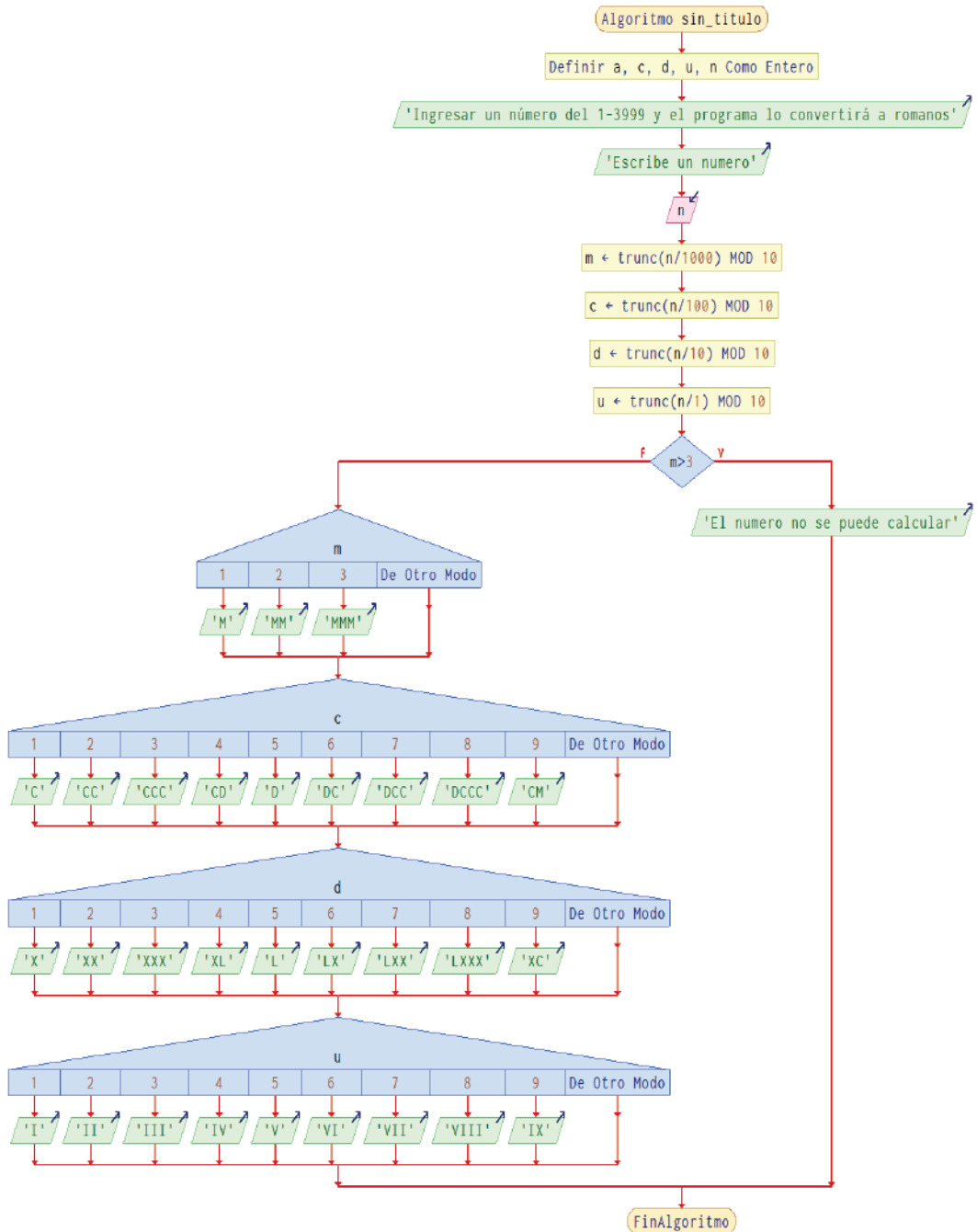
- 1: Escribir "I"
- 2: Escribir "II"
- 3: Escribir "III"
- 4: Escribir "IV"
- 5: Escribir "V"
- 6: Escribir "VI"
- 7: Escribir "VII"
- 8: Escribir "VIII"
- 9: Escribir "IX"

FinSegun

FinSi

FinAlgoritmo

3.9.2. Diagrama de flujo



3.9.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio9Bloque3 {

    public static void main(String[] args) {

        int numero, unidades, decenas, centenas, millares;

        Scanner entrada = new Scanner(System.in);

        System.out.println("Digite el número, puede estar entre 1 y 3000:");
        numero = entrada.nextInt();
        //4224

        if (numero > 0 && numero <= 3000){
            unidades = numero % 10;
            numero /= 10;
            decenas = numero % 10;
            numero /= 10;
            centenas = numero % 10;
            numero /= 10;
            millares = numero % 10;

            switch (millares){
                case 1:
                    System.out.print("M");
                    break;
            }
        }
    }
}
```

```
case 2:
    System.out.print("MM");
    break;
case 3:
    System.out.print("MMM");
    break;
}
switch (centenas){
    case 1:
        System.out.print("C");
        break;
    case 2:
        System.out.print("CC");
        break;
    case 3:
        System.out.print("CCC");
        break;
    case 4:
        System.out.print("CD");
        break;
    case 5:
        System.out.print("D");
        break;
    case 6:
        System.out.print("DC");
        break;
```

```
case 7:
    System.out.print("DCC");
    break;

case 8:
    System.out.print("DCCC");
    break;

case 9:
    System.out.print("CM");
    break;
}

switch (decenas){
    case 1:
        System.out.print("X");
        break;

    case 2:
        System.out.print("XX");
        break;

    case 3:
        System.out.print("XXX");
        break;

    case 4:
        System.out.print("XL");
        break;

    case 5:
        System.out.print("L");
        break;
```

```
        case 6:
            System.out.print("LX");
            break;

        case 7:
            System.out.print("LXX");
            break;

        case 8:
            System.out.print("LXXX");
            break;

        case 9:
            System.out.print("XC");
            break;
    }
    switch (unidades){
        case 1:
            System.out.print("I");
            break;

        case 2:
            System.out.print("II");
            break;

        case 3:
            System.out.print("III");
            break;

        case 4:
            System.out.print("IV");
            break;

        case 5:
            System.out.print("V");
            break;
```

```
        case 6:
            System.out.print("VI");
            break;

        case 7:
            System.out.print("VII");
            break;

        case 8:
            System.out.print("VIII");
            break;

        case 9:
            System.out.print("IX");
            break;
    }

} else if (numero == 0){
    System.out.println("No puede convertir 0 a romanos");
} else{
    System.out.println("Fuera del límite, espere una futura actualización");
}

}

}
```


3.10.

Mostrar los meses del año, pidiéndole al usuario un número entre (1-12), y mostrar el mes al que corresponde.

3.10.1. Código Pseint

Algoritmo Meses

Definir numero Como Entero

Escribir 'Ingrese un número entre 1 y 12:'

Leer numero

Segun numero Hacer

1:

Escribir 'Enero'

2:

Escribir 'Febrero'

3:

Escribir 'Marzo'

4:

Escribir 'Abril'

5:

Escribir 'Mayo'

6:

'Junio'

7:

Escribir 'Julio'

8:

Escribir 'Agosto'

9:

Escribir 'Septiembre'

10:

Escribir 'Octubre'

11:

Escribir 'Noviembre'

12:

Escribir 'Diciembre'

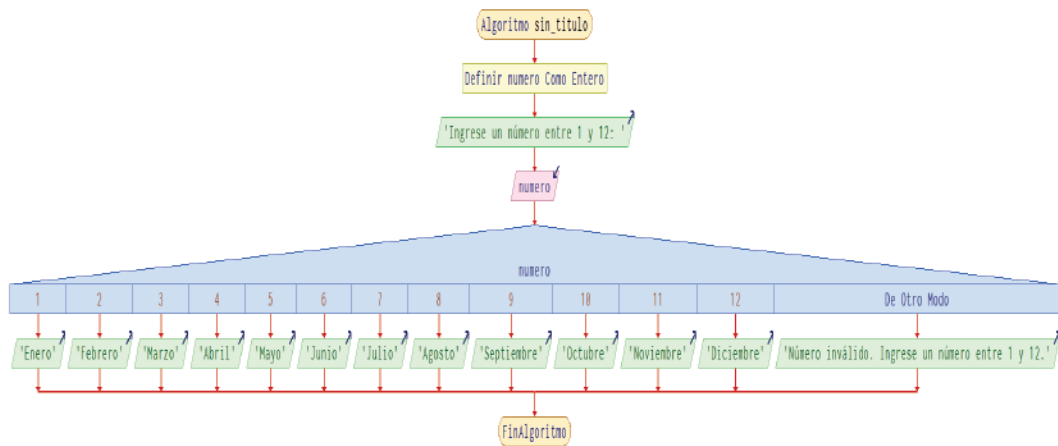
De Otro Modo:

Escribir 'Número inválido. Ingrese un número entre 1 y 12.'

FinSegun

FinAlgoritmo

3.10.2. Diagrama de flujo



3.10.3. Adaptación a Java

```

import java.util.Scanner;

public class MostrarMes {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese un número entre 1 y 12:");

        int numero = scanner.nextInt();

        String mes = "";

        switch (numero) {

            case 1:
                mes = "Enero";
                break;

            case 2:
                mes = "Febrero";
                break;

            case 3:
                mes = "Marzo";
                break;
        }
    }
}
  
```

```
case 4:
    mes = "Abril";
    break;

case 5:
    mes = "Mayo";
    break;

case 6:
    mes = "Junio";
    break;

case 7:
    mes = "Julio";
    break;

case 8:
    mes = "Agosto";
    break;

case 9:
    mes = "Septiembre";
    break;

case 10:
    mes = "Octubre";
    break;

case 11:
    mes = "Noviembre";
    break;

case 12:
    mes = "Diciembre";
    break;

default:
    System.out.println("Número inválido. El número debe estar entre 1 y
12.");
    return;
}
System.out.println("El número ingresado corresponde al mes de " + mes + ".");
}
```

3.11.

Hacer un programa que simule un cajero automático con un saldo inicial de 1000 dólares.

3.11.1. Código Pseint

Algoritmo B3_Ejercicio_11

Escribir "El programa simula un cajero automático con un saldo inicial de 1000 Dólares"

Definir saldoInicial, opcion Como Entero

saldoInicial <- 1000

Escribir "Bienvenido al Cajero Automático"

Escribir "Saldo disponible:", saldoInicial, " dólares"

Escribir ""

Escribir "Seleccione una opción:"

Escribir "1. Retirar dinero"

Escribir "2. Depositar dinero"

Escribir "3. Salir"

Leer opcion

Segun opcion Hacer

Caso 1:

Escribir "Ingrese la cantidad a retirar:"

Definir cantidadRetirar Como Entero

Leer cantidadRetirar

Si cantidadRetirar <= saldoInicial Entonces

saldolnicial <- saldolnicial - cantidadRetirar

Escribir "Retiro exitoso. Saldo actual:", saldolnicial, " dólares"

Sino

Escribir "Saldo insuficiente. No se puede realizar el retiro."

FinSi

Caso 2:

Escribir "Ingrese la cantidad a depositar:"

Definir cantidadDepositara Como Entero

Leer cantidadDepositara

saldolnicial <- saldolnicial + cantidadDepositara

Escribir "Depósito exitoso. Saldo actual:", saldolnicial, " dólares"

Caso 3:

Escribir "Gracias por utilizar el Cajero Automático"

Escribir "Saldo final:", saldolnicial, " dólares"

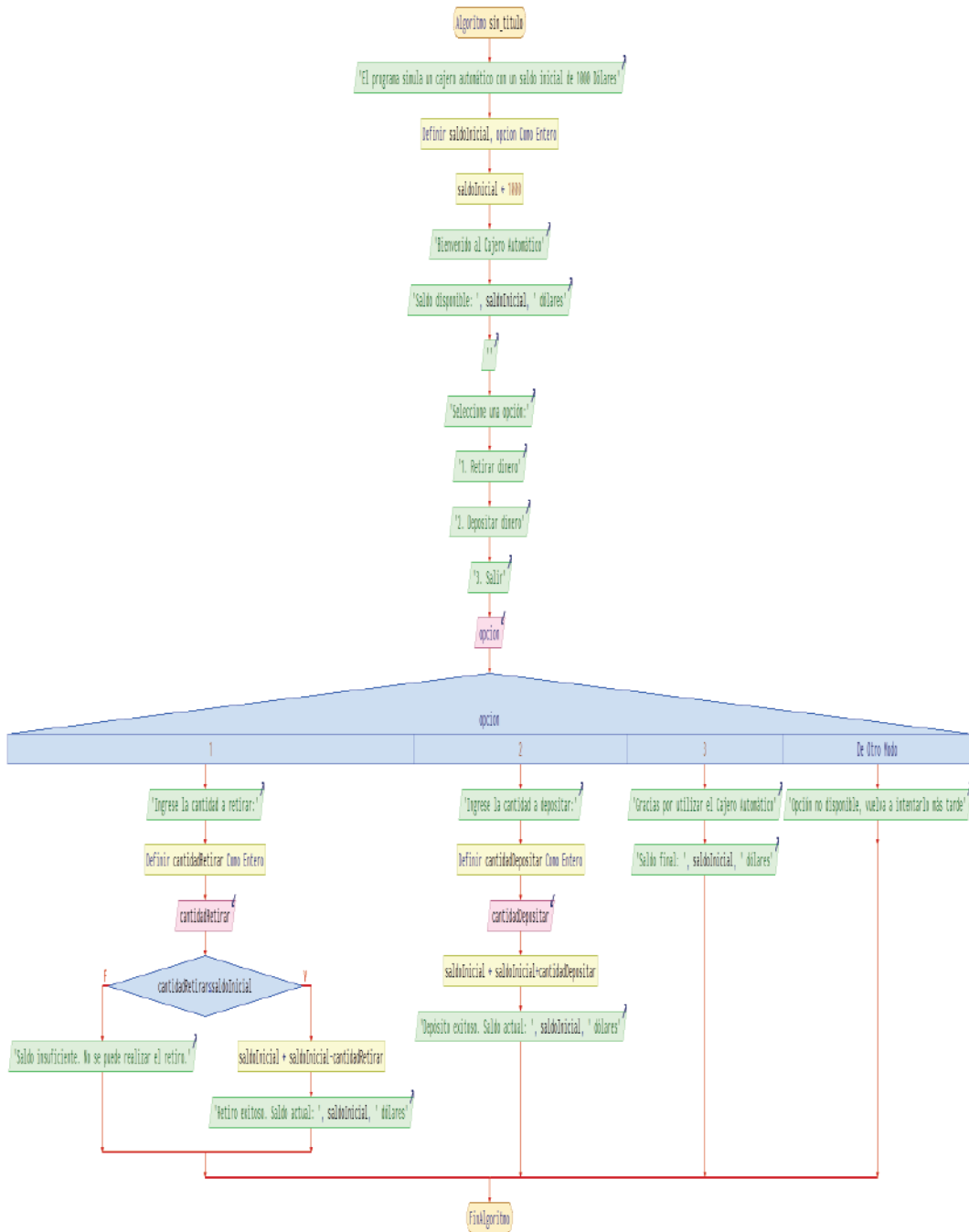
De Otro Modo:

Escribir "Opción no disponible, vuelva a intentarlo más tarde"

FinSegun

FinAlgoritmo

3.11.2. Diagrama de flujo



3.11.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio1Bloque3 {
    public static void main(String[] args) {

        int opcion;
        double dinero = 1000;
        double retiro, deposito;

        Scanner entrada = new Scanner(System.in);
        System.out.println("      Bienvenido al CAJERO      ");
        System.out.println("      ¿Qué Operación desea realizar?      ");
        System.out.println("1. retirar  2. Depositar  3. Consultar saldo");
        opcion = entrada.nextInt();
        if (opcion > 0 && opcion <= 3){
            switch (opcion){
                case 1:
                    System.out.println("Ingrese el monto a retirar:");
                    retiro = entrada.nextDouble();
                    if (retiro < dinero){
                        System.out.println("Su saldo actual es de " + (dinero - retiro) + " dólares");
                    } else{
                        System.out.println("Excede la cantidad que dispone, cerrando programa");
                    }
                    break;
                case 2:
                    System.out.println("Ingrese el monto a depositar:");
                    deposito = entrada.nextDouble();
                    System.out.println("Su saldo actual es ahora de " + (dinero + deposito) + " dolares");
                    break;
                case 3:
                    System.out.println("Su saldo actual es de " + dinero + " dólares");
                }
            } else{
                System.out.println("Opción no válida");
            }
        }
    }
}
```

3.12.

Hacer un menú que considere las siguientes opciones: Caso 1: Cubo de un numero Caso 2: Numero par o impar Case 3: salir.

3.12.1. Código Pseint

Algoritmo MenúOpciones

Definir opcion Como Entero

Repetir

Escribir "MENU DE OPCIONES"

Escribir "1. Calcular el cubo de un número"

Escribir "2. Determinar si un número es par o impar"

Escribir "3. Salir"

Escribir ""

Escribir "Ingrese una opción (1-3):"

Leer opcion

Segun opcion Hacer

Caso 1:

Definir numero, cubo Como Real

Escribir "Ingrese un número:"

Leer numero

cubo <- numero * numero * numero

Escribir "El cubo de ", numero, " es:", cubo

Escribir ""

Caso 2:

Definir numero Como Entero

Escribir "Ingrese un número:"

Leer numero

Si numero % 2 = 0 Entonces

Escribir "El número ", numero, " es par."

Sino

Escribir "El número ", numero, " es impar."

FinSi

Escribir ""

Caso 3:

Escribir "Saliendo del programa..."

De Otro Modo:

Escribir "Opción inválida. Ingrese una opción válida (1-3)."

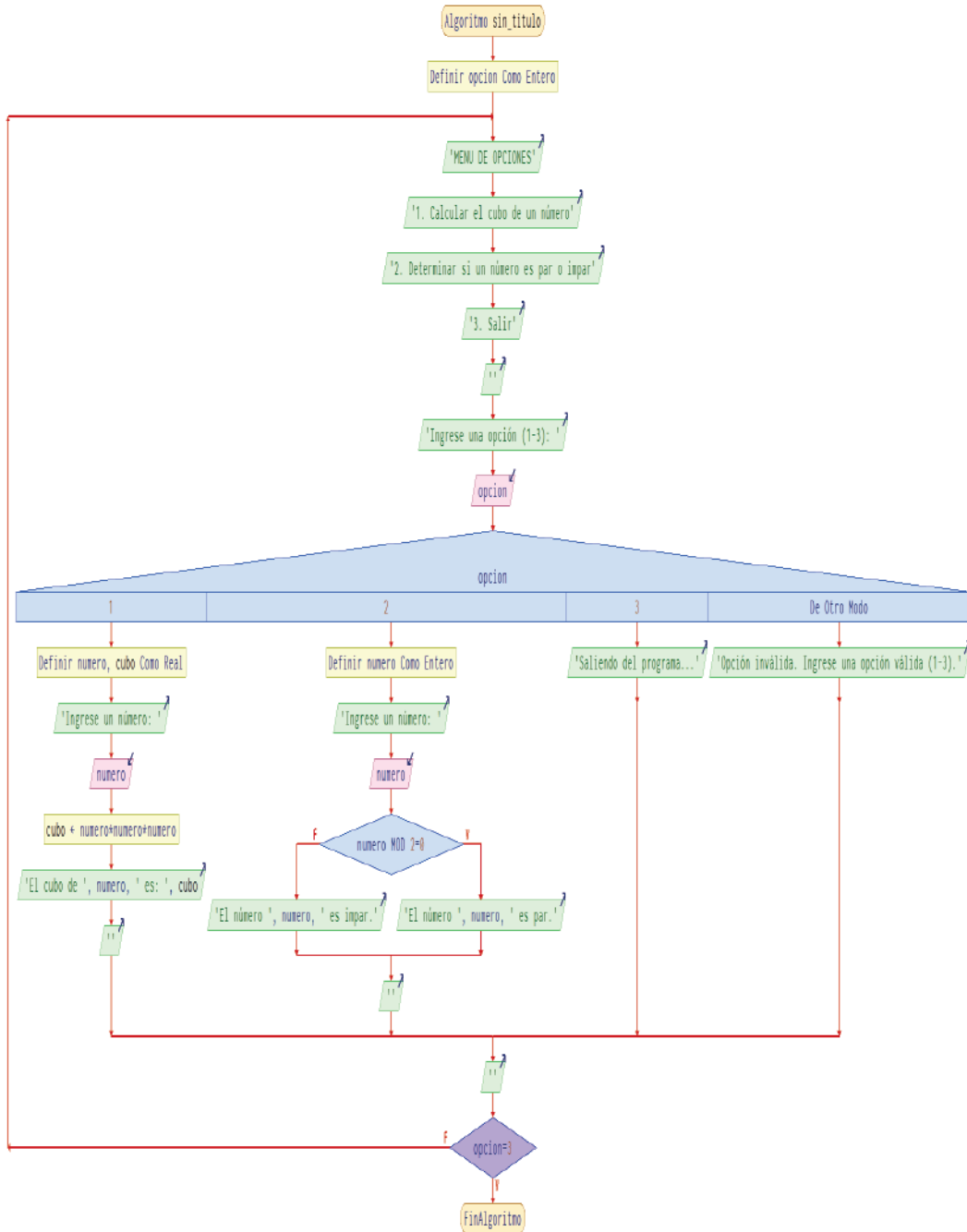
FinSegun

Escribir ""

Hasta Que opcion = 3

FinAlgoritmo

3.12.2. Diagrama de flujo



3.12.3. Adaptación a Java

```
import java.util.Scanner;

public class MenuOpciones {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int opcion;

        do {

            System.out.println("MENU DE OPCIONES");

            System.out.println("1. Cubo de un número");

            System.out.println("2. Número par o impar");

            System.out.println("3. Salir");

            System.out.print("Ingrese una opción:");

            opcion = scanner.nextInt();

            switch (opcion) {

                case 1:

                    System.out.print("Ingrese un número:");

                    int numero = scanner.nextInt();

                    int cubo = numero * numero * numero;

                    System.out.println("El cubo de " + numero + " es " + cubo);

                    break;

                case 2:

                    System.out.print("Ingrese un número:");

                    int num = scanner.nextInt();

                    if (num % 2 == 0) {

                        System.out.println("El número " + num + " es par");

                    }

                }

            }

        } while (opcion != 3);

    }

}
```

```
        } else {  
            System.out.println("El número " + num + " es impar");  
        }  
        break;  
    case 3:  
        System.out.println("Saliendo del programa..");  
        break;  
    default:  
        System.out.println("Opción inválida. Ingrese una opción válida.");  
        break;  
    }  
    System.out.println(); // Agregar una línea en blanco para separar las itera-  
ciones  
} while (opcion != 3);  
System.out.println("Fin del programa");  
}  
}
```

Bloque 04: Estructuras repetitivas

4.1.

Realice un programa que solicite de la entrada estándar un entero del 1 al 10 y muestre en la salida estándar su tabla de multiplicar.

4.1.1. Código Pseint

Algoritmo B4_Ejercicio_1

```
Definir nro,resultado Como Entero
Escribir 'Ingresar un número del 1-10 y el programa mostrará su tabla de
multiplicar'

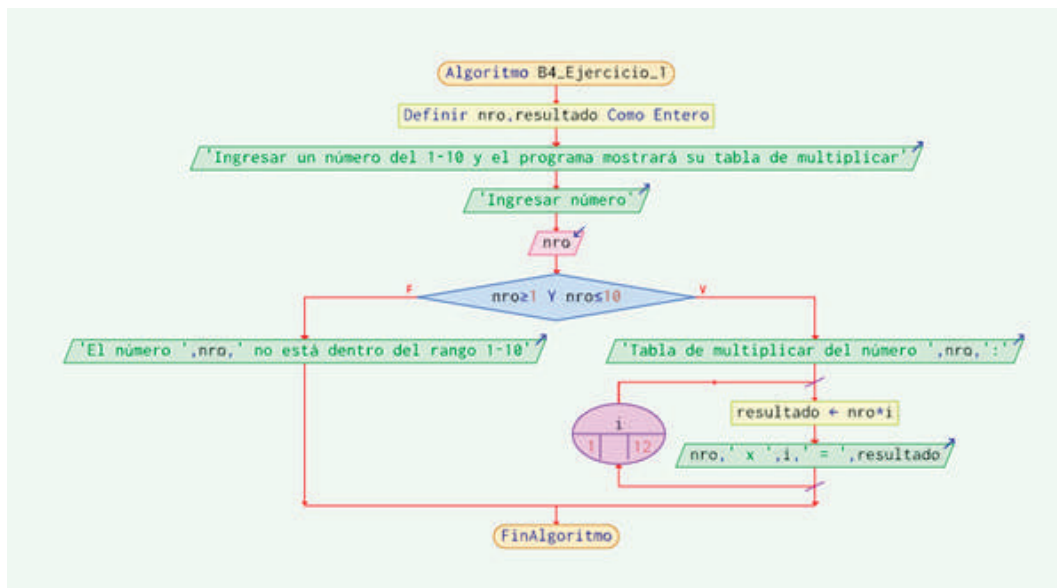
Escribir 'Ingresar número'
Leer nro

Si nro>=1 Y nro<=10 Entonces
  Escribir 'Tabla de multiplicar del número ',nro,':'
  Para i<-1 Hasta 12 Hacer
    resultado <- nro*i
    Escribir nro,' x ',i,' = ',resultado

  FinPara
SiNo
  Escribir 'El número ',nro,' no está dentro del rango 1-10'
FinSi
```

FinAlgoritmo

4.1.2. Diagrama de flujo



4.1.3. Adaptación a Java

```

import java.util.Scanner;

public class Mavenproject1 {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        int numero, i;

        System.out.println("ingrese su numero del 1 al 10");

        numero = entrada.nextInt();

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

            System.out.println(numero+ "x" + i+ "=" +(numero*i));

        }

    }

}
  
```

4.2.

Realice un programa que lea de la entrada estándar números hasta que se introduzca un cero. En ese momento el programa debe terminar y mostrar en la salida estándar el número de valores mayores que cero leídos.

4.2.1. Código Pseint

Algoritmo EntradaNúmeros

Definir numero, contadorPositivos Como Entero

contadorPositivos <- 0

Escribir "Ingrese números (ingrese 0 para terminar):"

Leer numero

Mientras numero <> 0 Hacer

Si numero > 0 Entonces

 contadorPositivos <- contadorPositivos + 1

FinSi

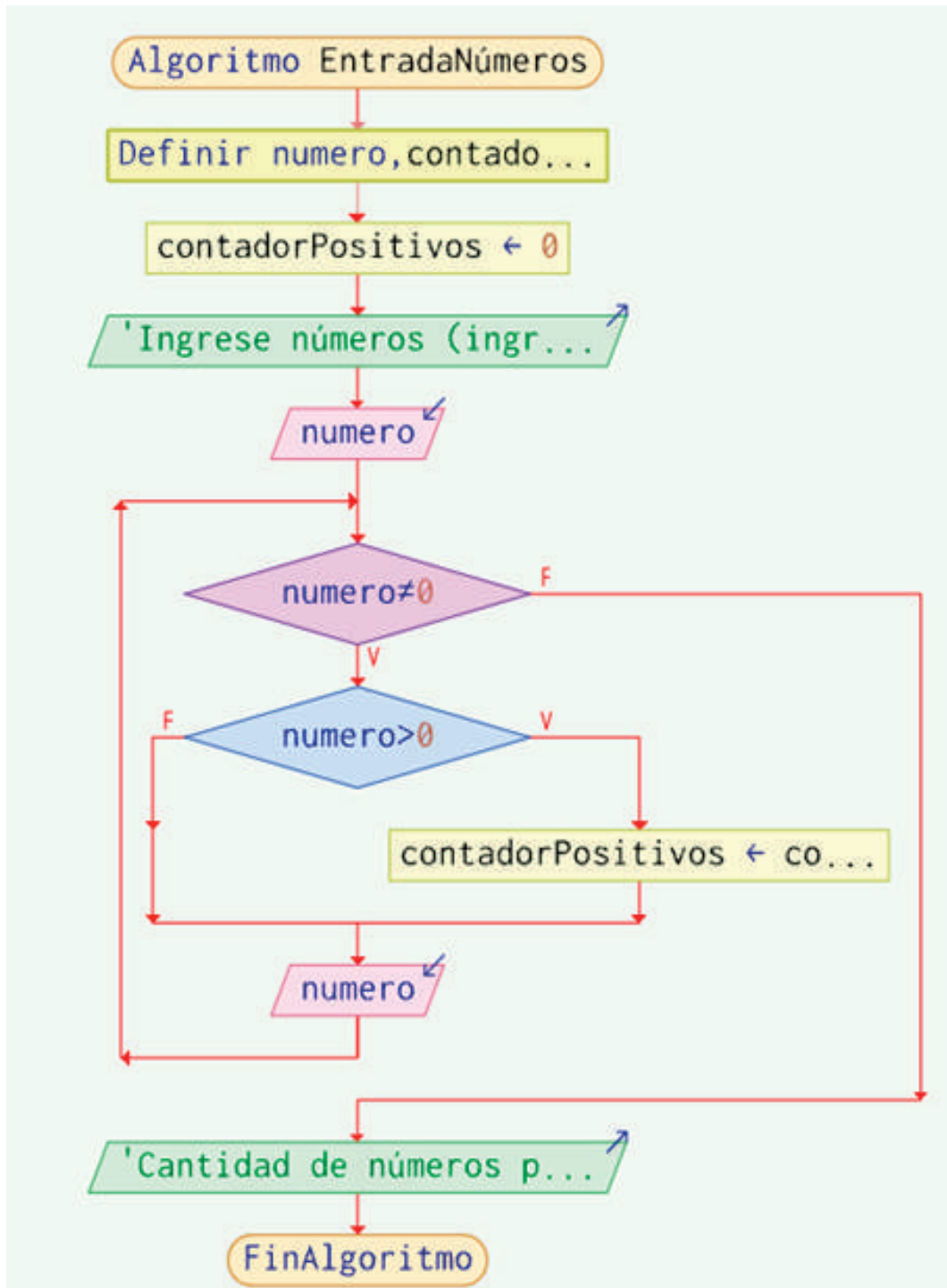
Leer numero

FinMientras

Escribir "Cantidad de números positivos leídos:", contadorPositivos

FinAlgoritmo

4.2.2. Diagrama de flujo



4.2.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio2Bloque4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int numero;
        int contador = 0;
        do {
            System.out.print("Ingrese un número (0 para salir):");
            numero = scanner.nextInt();
            if (numero > 0) {
                contador++;
            }
        } while (numero != 0);
        System.out.println("La cantidad de números mayores que cero ingresados es:"
            + contador);
    }
}
```

4.3.

Realice un programa que calcule y muestre en la salida estándar la suma de los cuadrados de los 10 primeros enteros mayores que cero.

4.3.1. Código Pseint

Algoritmo B4_Ejercicio_3

Escribir “El programa calcula y muestra en la salida estándar la suma de los cuadrados de los 10 primeros enteros mayores que cero”

Definir suma, numero, cuadrado Como Entero

suma <- 0

Para numero <- 1 Hasta 10 Hacer

cuadrado <- numero ^ 2

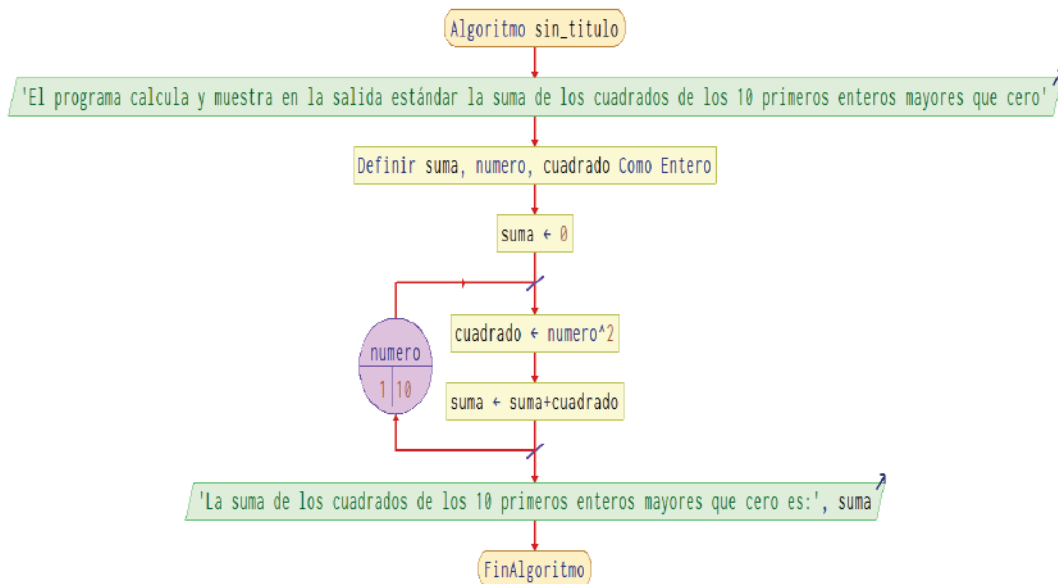
suma <- suma + cuadrado

FinPara

Escribir “La suma de los cuadrados de los 10 primeros enteros mayores que cero es:”, suma

FinAlgoritmo

4.3.2. Diagrama de flujo



4.3.3. Adaptación a Java

```
import java.util.Scanner;
public class Mavenproject1 {
    public static void main(String[] args) {
        int suma_cuadrados = 0;
        for (int i = 1; i <= 10; i++) {
            int cuadrado = i * i;
            suma_cuadrados += cuadrado;
        }
        System.out.println("La suma de los cuadrados de los 10 primeros enteros ma-
        yores que cero es:" + suma_cuadrados);
    }
}
```

44.

Escriba un programa que tome cada 4 horas la temperatura exterior, leyéndola durante un período de 24 horas. Es decir, debe leer 6 temperaturas. Calcule la temperatura media del día, la temperatura más alta y la más baja.

44.1. Código Pseint

Algoritmo temperaturasSemana

```
FILAS<-5
COLUMNAS<-7
LUNES<-0
MARTES<-1
```

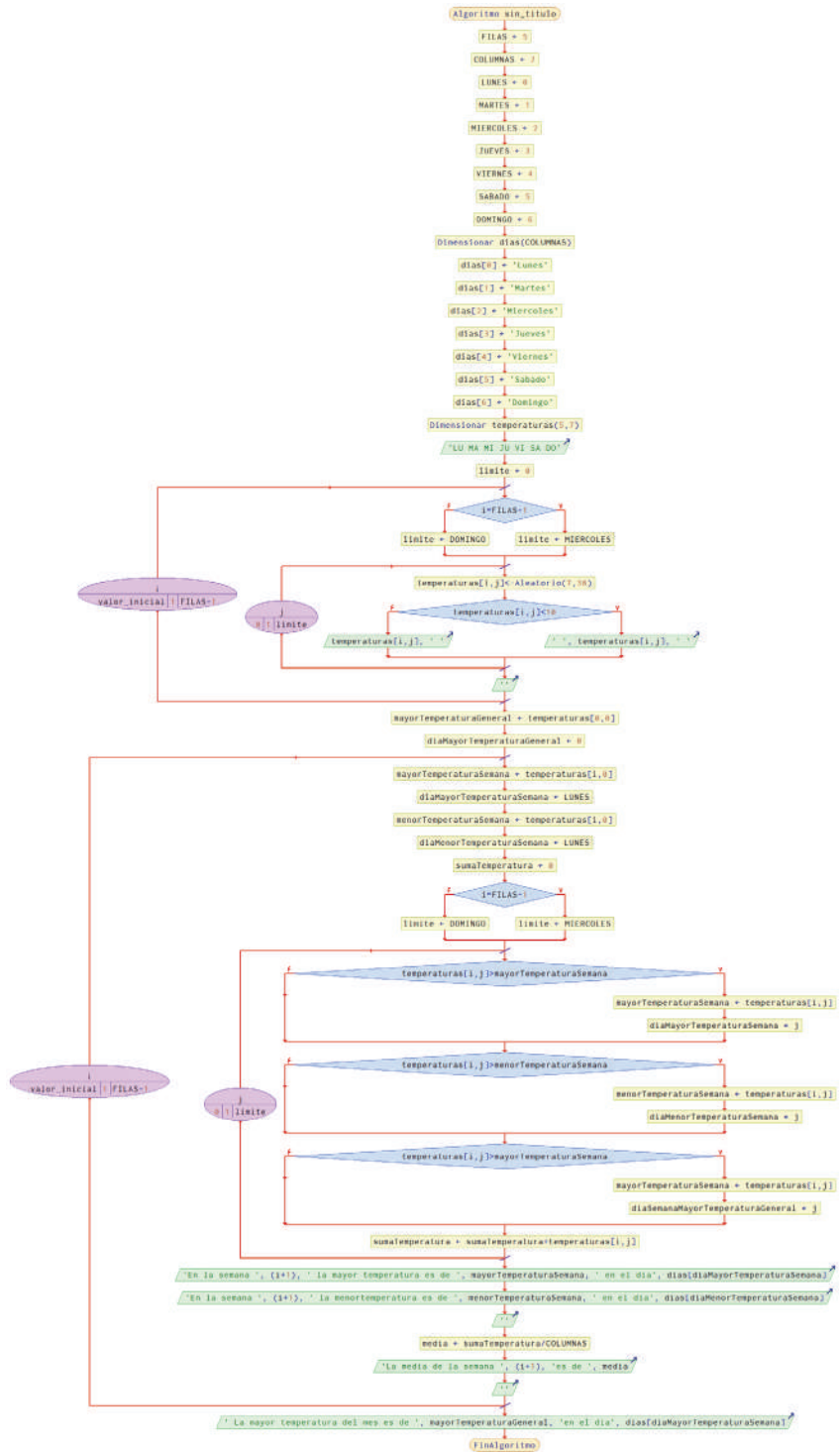
```
MIERCOLES<-2
JUEVES<-3
VIERNES<-4
SABADO<-5
DOMINGO<-6
Dimension dias(COLUMNAS)
dias(0)<-"Lunes"
dias(1)<-"Martes"
dias(2)<-"Miercoles"
dias(3)<-"Jueves"
dias(4)<-"Viernes"
dias(5)<-"Sabado"
dias(6)<-"Domingo"
Dimension temperaturas[5,7];
Escribir "LU MA MI JU VI SA DO";
limite <- 0
Para i<-valor_inicial Hasta FILAS-I Con Paso 1 Hacer
  Si i=FILAS-I Entonces
    limite <- MIERCOLES
  SiNo
    limite <- DOMINGO
  Fin Si
Para j<-0 Hasta limite Con Paso 1 Hacer
  temperaturas(i,j) <- Aleatorio(7,38)
  Si temperaturas(i,j)<10 Entonces
```

```
Escribir "" ,temperaturas(i,j) "" Sin Saltar
SiNo
Escribir temperaturas(i,j) "" Sin Saltar
Fin Si
Fin Para
Escribir ""
Fin Para
mayorTemperaturaGeneral <- temperaturas(0,0)
diaMayorTemperaturaGeneral <- 0
Para i<-valor_inicial Hasta FILAS-1 Con Paso 1 Hacer
    mayorTemperaturaSemana <- temperaturas(i,0)
    diaMayorTemperaturaSemana <- LUNES
    menorTemperaturaSemana <- temperaturas(i,0)
    diaMenorTemperaturaSemana <- LUNES
    sumaTemperatura <- 0
    Si i=FILAS-1 Entonces
        limite <- MIERCOLES
        SiNo
            limite <- DOMINGO
        Fin Si
        Para j<-0 Hasta limite Con Paso 1 Hacer
            Si temperaturas(i,j)>mayorTemperaturaSemana Entonces
                mayorTemperaturaSemana <- temperaturas(i,j)
                diaMayorTemperaturaSemana <- j
            Fin Si
```

```
Si temperaturas(i,j)>menorTemperaturaSemana Entonces
menorTemperaturaSemana <- temperaturas(i,j)
diaMenorTemperaturaSemana <- j
Fin Si
Si temperaturas(i,j)>mayorTemperaturaSemana Entonces
mayorTemperaturaSemana <- temperaturas(i,j)
diaSemanaMayorTemperaturaGeneral <- j
Fin Si
sumaTemperatura<-sumaTemperatura + temperaturas(i,j)
Fin Para
Escribir "En la semana ", (i+1) , " la mayor temperatura es de ",mayorTemperaturaSemana, " en el dia" dias(diaMayorTemperaturaSemana)
Escribir "En la semana ", (i+1) , " la menor temperatura es de ",menorTemperaturaSemana, " en el dia" dias(diaMenorTemperaturaSemana)
Escribir ""
media<- sumaTemperatura / COLUMNAS
Escribir "La media de la semana ",(i+1), "es de ",media
Escribir ""
Fin Para
Escribir " La mayor temperatura del mes es de ",mayorTemperaturaGeneral, " en el dia" dias(diaMayorTemperaturaSemana)
```

FinAlgoritmo

44.2. Diagrama de flujo



44.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio4Bloque4{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int totalTemperaturas = 6;

        int intervaloHoras = 4;

        double temperatura;
        double sumaTemperaturas = 0;
        double temperaturaMaxima = Double.MIN_VALUE;
        double temperaturaMinima = Double.MAX_VALUE;
        for (int i = 0; i < totalTemperaturas; i++) {

            int horasTranscurridas = i * intervaloHoras;

            System.out.print("Ingrese la temperatura exterior después de " + horasTranscu-
                rridas + " horas:");

            temperatura = scanner.nextDouble();

            sumaTemperaturas += temperatura;

            if (temperatura < temperaturaMinima) {

                temperaturaMinima = temperatura;

            }

        }

        double temperaturaMedia = sumaTemperaturas / totalTemperaturas;

        System.out.println("Temperatura media del día:" + temperaturaMedia);
        System.out.println("Temperatura más alta:" + temperaturaMaxima);
        System.out.println("Temperatura más baja:" + temperaturaMinima);

    }

}
```


4.5.

Escriba un programa que lea valores enteros hasta que se introduzca un valor en el rango [20-30] o se introduzca el valor 0. El programa debe entregar la suma de los valores mayores a 0 introducidos.

4.5.1. Código Pseint

Algoritmo B4_Ejercicio_5

Definir valor, suma Como Entero

suma <- 0

Escribir “El programa lee valores enteros hasta que se introduzca un valor en el rango [20-30] o se introduzca el valor 0. El programa entrega la suma de los valores mayores a 0 introducidos.”

Escribir “”

Hacer

Escribir “Ingrese un valor entero (0 para terminar):”

Leer valor

Si valor > 0 Entonces

suma <- suma + valor

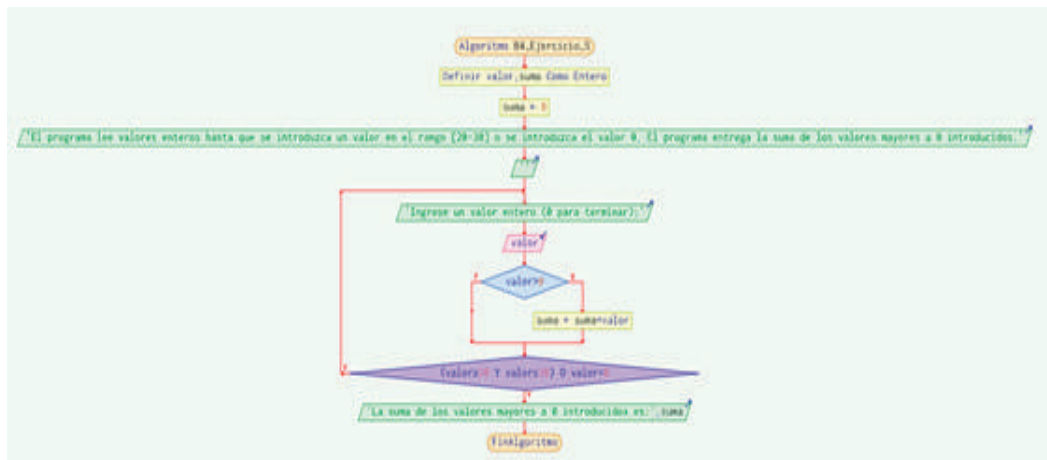
FinSi

Hasta Que (valor >= 20 Y valor <= 30) O valor = 0

Escribir “La suma de los valores mayores a 0 introducidos es:”, suma

FinAlgoritmo

4.5.2. Diagrama de flujo



4.5.3. Adaptación a Java

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

```
public class Main{
```

```
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        int entero = 0, suma = 0, band;
```

```
        do{
```

```
            band = 0;
```

```
            suma += entero;
```

```
            System.out.println("Digite un número:");
```

```
            entero = entrada.nextInt();
```

```
            if (entero != 0){
```

```
                band += 1;
```

```
            }
```

```
            if (entero < 20){
```

```
                band += 1;
```

```
            }
```

```
            if (entero > 30){
```

```
                band +=1;
```

```
            }
```

```
        } while (band >= 2);
```

```
        System.out.println("La suma de los enteros mayores que cero y fuera del
rango [20-30] es:" + suma);
```

```
    }
```

```
}
```

4.6.

Escriba un programa que calcule x^y , donde tanto x como y son enteros positivos, sin utilizar la función `pow()`.

4.6.1. Código Pseint

Algoritmo xPy

Definir $n, p, potencia$ Como Entero

Escribir "Escribe un número"

Leer n

Escribir "Escribe la potencia a realizar"

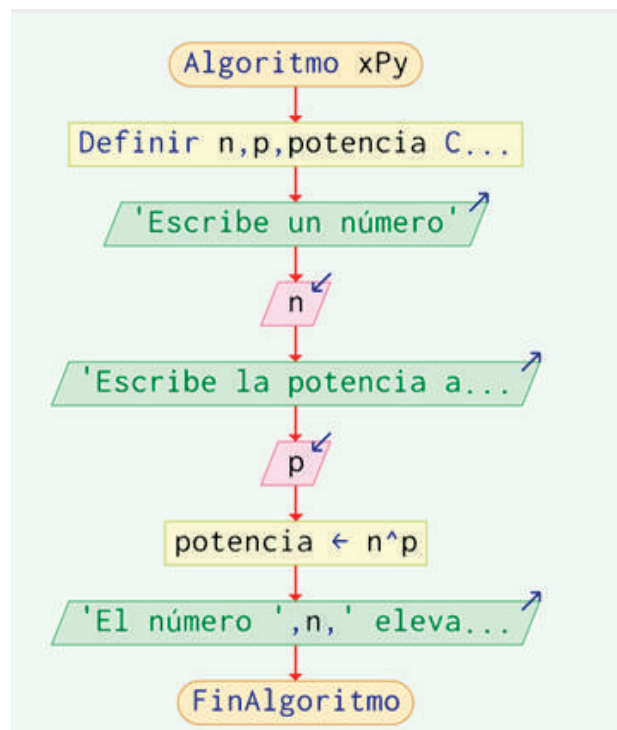
Leer p

$potencia = n^p$

Escribir "El número ' n ,' elevado a la potencia ' p ,' es: ' $potencia$ '"

FinAlgoritmo

4.6.2. Diagrama de flujo



4.6.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio6Bloque4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese la base (x):");
        int x = scanner.nextInt();
        System.out.print("Ingrese el exponente (y):");
        int y = scanner.nextInt();
        int resultado = 1;
        for (int i = 0; i < y; i++) {
            resultado *= x;
        }
        System.out.println(x + "^" + y + " = " + resultado);
    }
}
```

4.7.

Escriba un programa que calcule el valor de: $1+2+3+\dots+n$

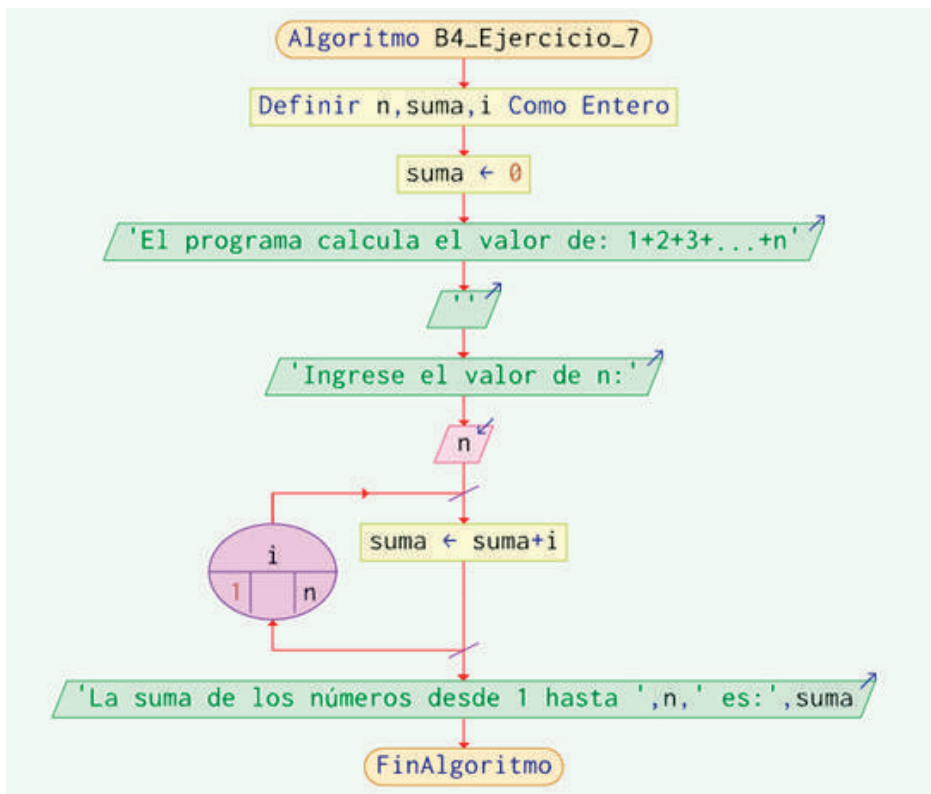
4.7.1. Código Pseint

Algoritmo B4_Ejercicio_7

Definir n, suma, i Como Entero
 suma \leftarrow 0
 Escribir "El programa calcula el valor de: 1+2+3+...+n"
 Escribir ""
 Escribir "Ingrese el valor de n:"
 Leer n
 Para i \leftarrow 1 Hasta n Hacer
 suma \leftarrow suma + i
 FinPara
 Escribir "La suma de los números desde 1 hasta ", n, " es:", suma

FinAlgoritmo

4.7.2. Diagrama de flujo



4.7.3. Adaptación a Java

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner dt=new Scanner(System.in);

        int i=1, numero=0, lm;

        System.out.println("Ingrese limite:");

        lm=dt.nextInt();

        do{

            System.out.print(i+"");

            numero=numero+i;

            i++;

        }while(i<=lm);

        System.out.println("la suma es "+numero);

    }

}
```

4.8.

Escriba un programa que calcule el valor de: $1+3+5+\dots+2n-1$

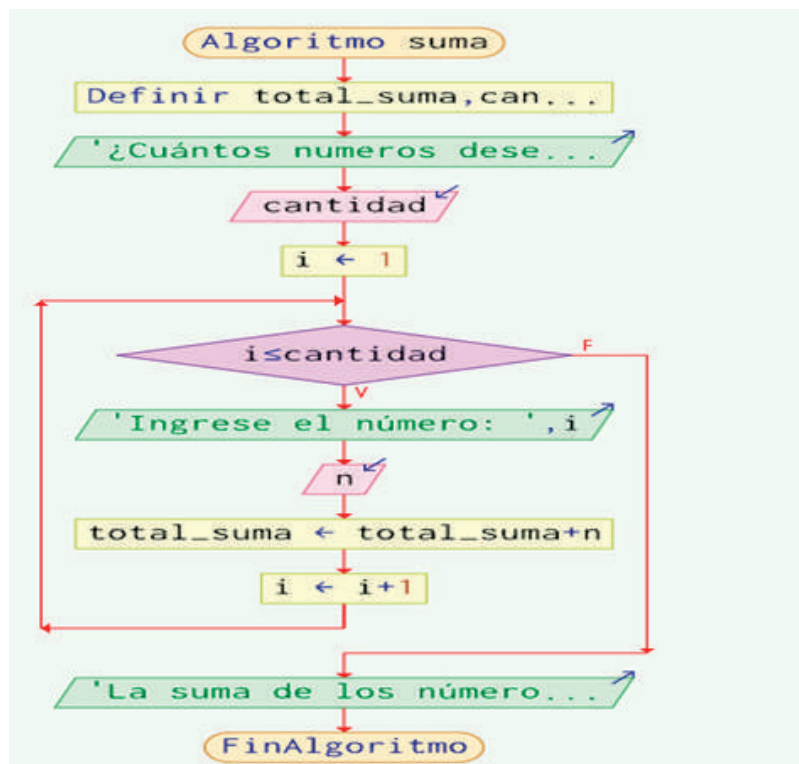
4.8.1. Código Pseint

Algoritmo suma

```
Definir total_suma, cantidad, i Como Entero
Escribir "¿Cuántos numeros desea sumar?"
leer cantidad
i<-1
Mientras i<=cantidad Hacer
  Escribir "Ingrese el número:",i
  leer n
  total_suma<-total_suma+n
  i<-i+1
Fin Mientras
Escribir "La suma de los números es:", total_suma
```

FinAlgoritmo

4.8.2. Diagrama de Flujo



4.8.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio8Bloque4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el valor de n:");
        int n = scanner.nextInt();
        int suma = 0;
        for (int i = 1; i <= 2 * n - 1; i += 2) {
            suma += i;
        }
        System.out.println("La suma de los números impares desde 1 hasta " + (2 * n - 1) + " es: " + suma);
    }
}
```

4.9.

Escriba un programa que calcule el valor de: $1*2*3*...*n$ (factorial)

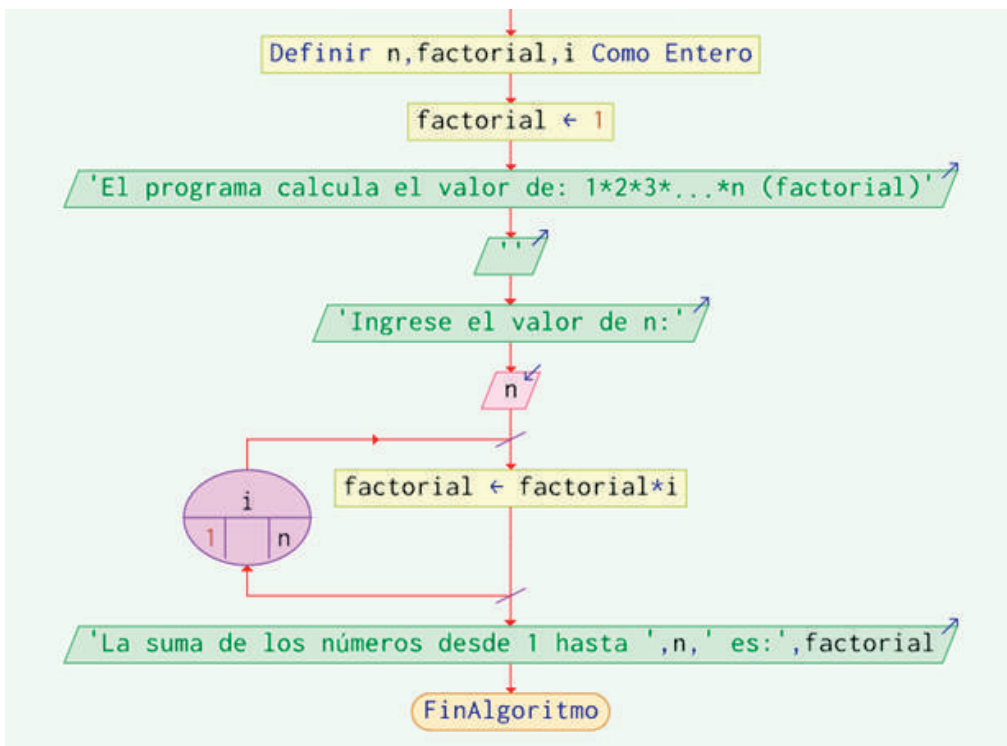
4.9.1. Código Pseint

Algoritmo B4_Ejercicio_9

```
Definir n, factorial, i Como Entero
factorial <- 1
Escribir "El programa calcula el valor de: 1*2*3*...*n (factorial)"
Escribir ""
Escribir "Ingrese el valor de n:"
Leer n
Para i <- 1 Hasta n Hacer
    factorial <- factorial * i
FinPara
Escribir "La suma de los números desde 1 hasta ", n, " es:", factorial
```

FinAlgoritmo

4.9.2. Diagrama de flujo



4.9.3. Adaptación a Java

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

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingresa un número entero positivo:");
        int n = scanner.nextInt();
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += 2 * i - 1;
        }
        System.out.println("La suma de los primeros " + n + " números impares es: " +
            sum);
    }
}
```

4.10.

Escriba un programa que calcule el valor de: $1!+2!+3!+\dots+n!$ (suma de factoriales).

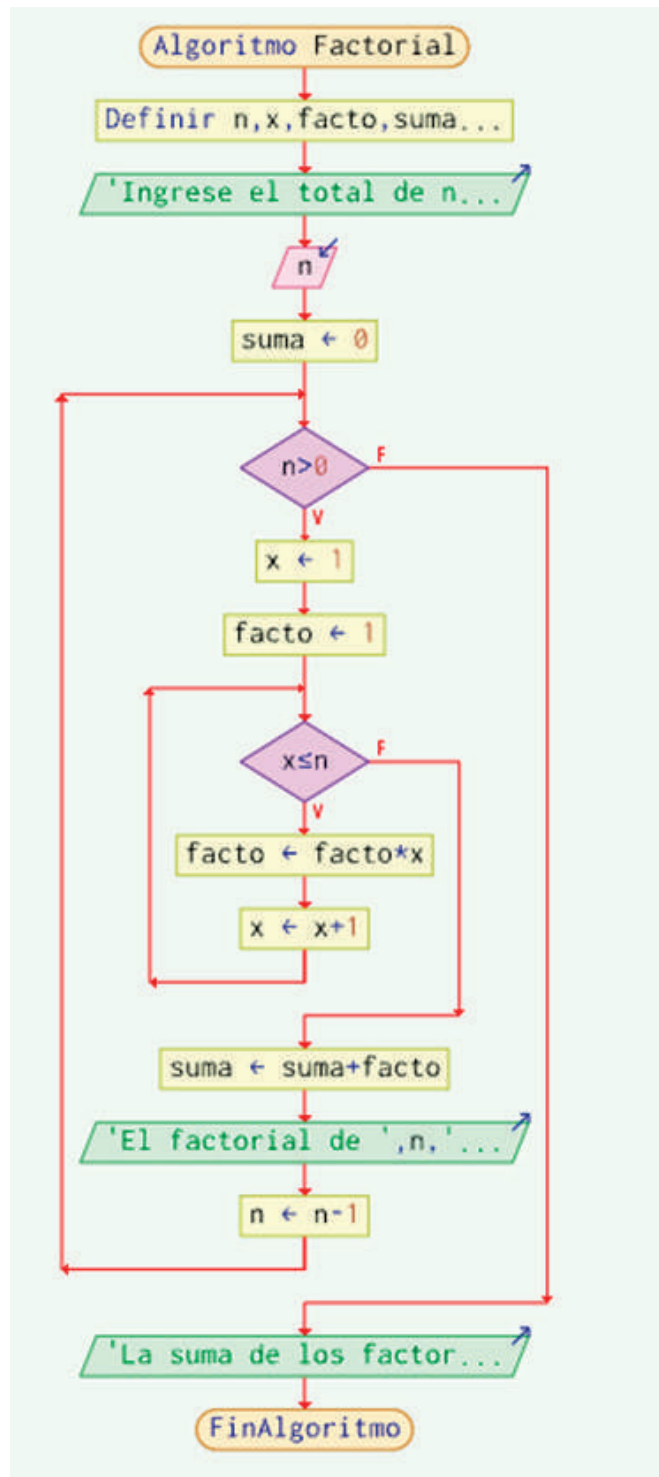
4.10.1. Código Pseint

Algoritmo Factorial

```
Definir n,x,facto,suma Como Entero
Escribir "Ingrese el total de número"
leer n
suma = 0
Mientras n > 0 Hacer
    x = 1
    facto = 1
    Mientras x <= n Hacer
        facto = facto * x
        x = x + 1
    FinMientras
    suma = suma + facto
    Escribir "El factorial de ",n," es:",facto
    n = n - 1
FinMientras
Escribir "La suma de los factoriales es:",suma
```

FinAlgoritmo

4.10.2. Diagrama de flujo



4.10.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio10Bloque4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el valor de n:");
        int n = scanner.nextInt();
        int suma = 0;
        for (int i = 1; i <= n; i++) {
            int factorial = 1;
            for (int j = 1; j <= i; j++) {
                factorial *= j;
            }
            suma += factorial;
        }
        System.out.println("La suma de factoriales desde 1 hasta " + n + " es: " + suma);
    }
}
```

4.11.

Escriba un programa que calcule el valor de: $2^1+2^2+2^3+\dots+2^n$

4.11.1. Código Pseint

Algoritmo B4_Ejercicio_11

Definir n, suma, i, potencia Como Entero

Escribir "El programa calcula el valor de: $2^1+2^2+2^3+\dots+2^n$ "

Escribir ""

suma <- 0

potencia <- 1

Escribir "Ingrese el valor de n:"

Leer n

Para i <- 1 Hasta n Hacer

 potencia <- potencia * 2

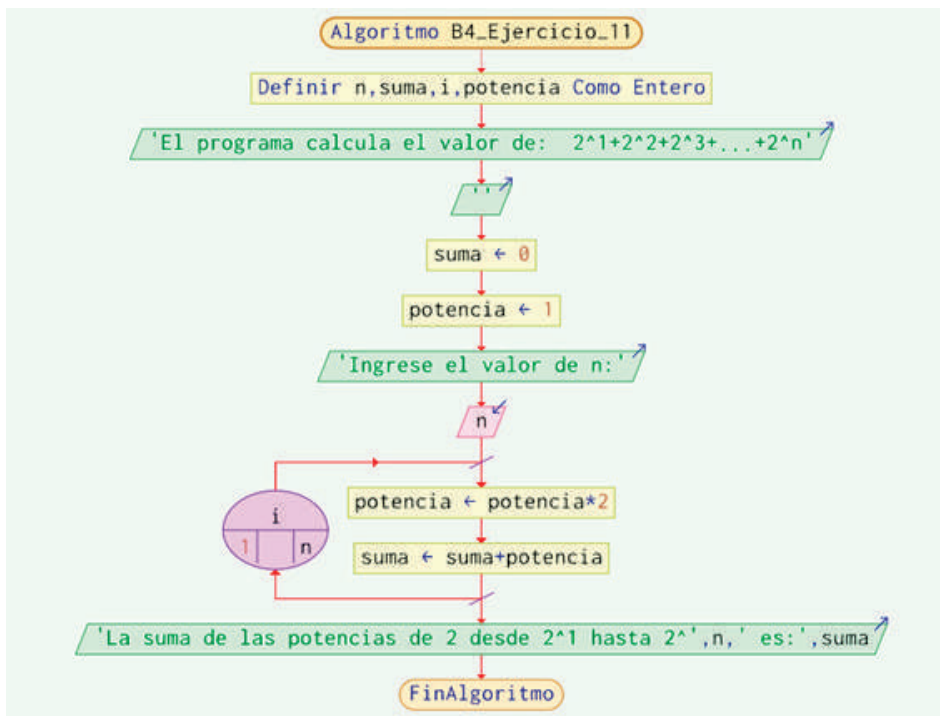
 suma <- suma + potencia

FinPara

Escribir "La suma de las potencias de 2 desde 2^1 hasta 2^n , n, " es:", suma

FinAlgoritmo

4.11.2. Diagrama de flujo



4.11.3. Adaptación a Java

```

import java.util.Scanner;

public class Main {

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

        System.out.println("Ingrese el valor de n:");
        int n = teclado.nextInt();

        int suma = 0;
        for (int i = 0; i <= n; i++) {
            suma += Math.pow(2, i);
        }
        System.out.println("La sumatoria es "+ suma);
    }
}
    
```

4.12.

Hacer un programa que calcule el resultado de la siguiente expresión: $1-2+3-4+5-6\dots n$

4.12.1. Código Pseint

Algoritmo CalculoExpresion

Definir n, resultado, i Como Entero

Escribir "Ingrese el valor de n:"

Leer n

resultado <- 0

i <- 1

Mientras i <= n Hacer

 Si $i \% 2 = 0$ Entonces

 resultado <- resultado - i

 Sino

 resultado <- resultado + i

 FinSi

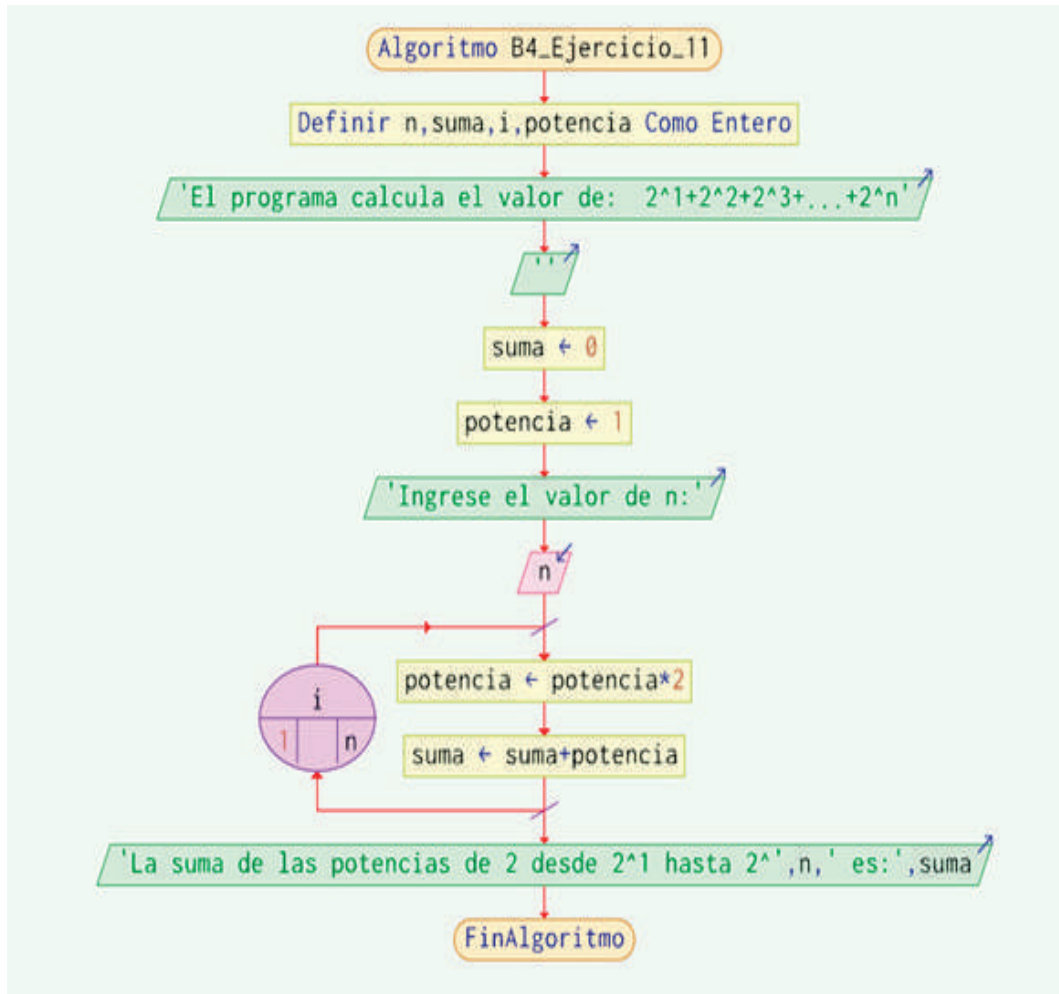
 i <- i + 1

FinMientras

Escribir "El resultado de la expresión es:", resultado

FinAlgoritmo

4.12.2. Diagrama de flujo



4.12.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio12Bloque4{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el valor de n:");
        int n = scanner.nextInt();
        int resultado = 0;
        int signo = 1;
        for (int i = 1; i <= n; i++) {
            resultado += signo * i;
            signo *= -1;
        }
        System.out.println("El resultado de la expresión es:" + resultado);
    }
}
```

4.13. Treceavo ejercicio

Hacer un que realice la serie fibonacci -> 1 1 2 3 5 8 13...n

4.13.1. Código Pseint

Algoritmo B4_Ejercicio_13

Definir n, a, b, c Como Entero

Escribir "El programa calcula la serie de fibonacci -> 1 1 2 3 5 8 13...n"

Escribir ""

Escribir "Ingrese el número de elementos de la serie Fibonacci:"

Leer n

a <- 1

b <- 1

Escribir a

Escribir b

Para i <- 3 Hasta n Hacer

 c <- a + b

 Escribir c

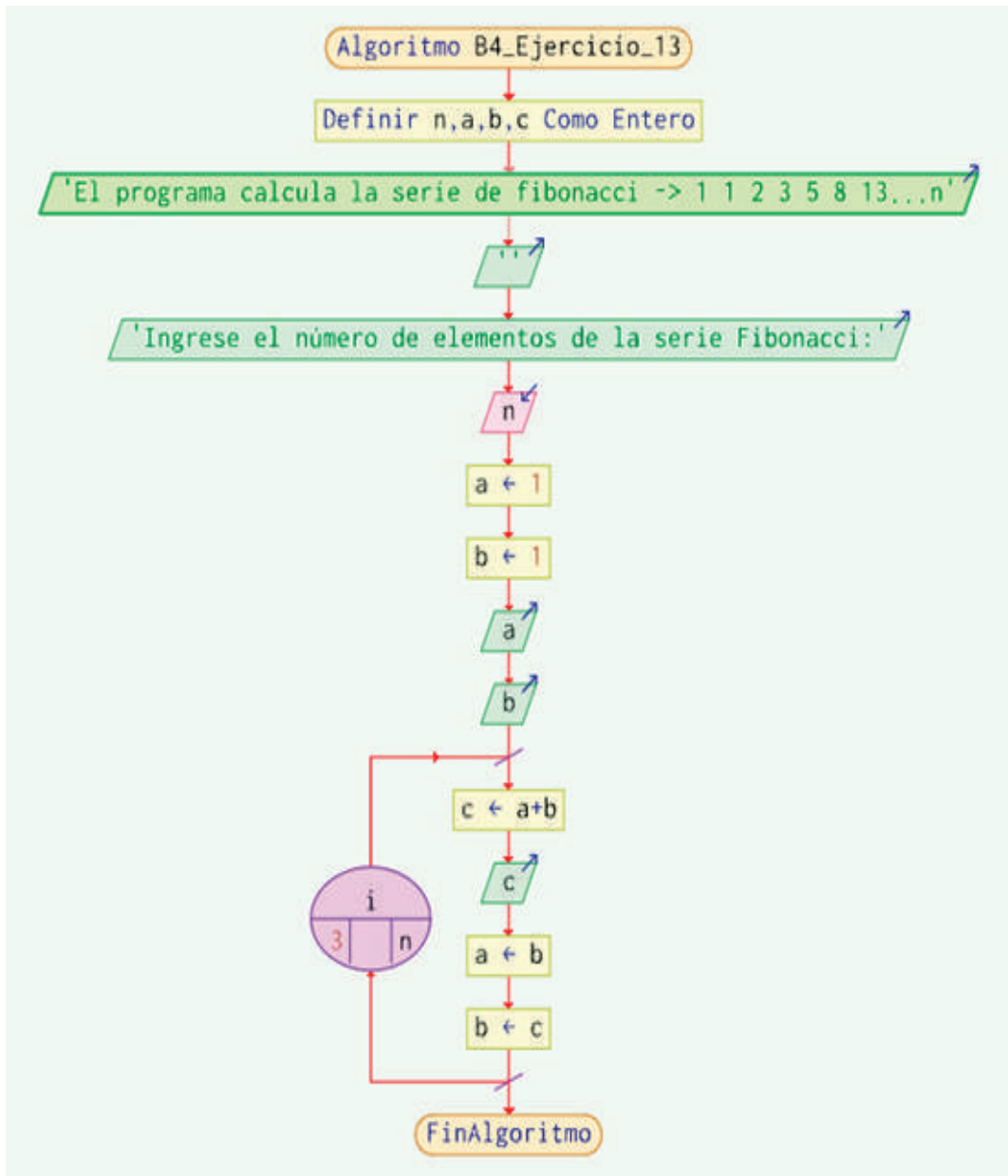
 a <- b

 b <- c

FinPara

FinAlgoritmo

4.13.2. Diagrama de flujo



4.13.3. Adaptación a Java

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        Scanner teclado=new Scanner(System.in);

        int a=0,b=1,c,n;

        System.out.println("Ingrese la cantidad de datos de la serie:");

        n =teclado.nextInt();

        System.out.println("La serie Fibonacci es:");

        for (int i = 0; i < n; i++) {

            System.out.println(a);

            c=a+b;

            a=b;

            b=c;

        }

    }

}
```

4.14.

En una clase de 5 alumnos se han realizado tres exámenes y se requiere determinar el número de:

- a) Alumnos que aprobaron todos los exámenes.
- b) Alumnos que aprobaron al menos un examen.
- c) Alumnos que aprobaron únicamente el último examen.

4.14.1. Código Pseint

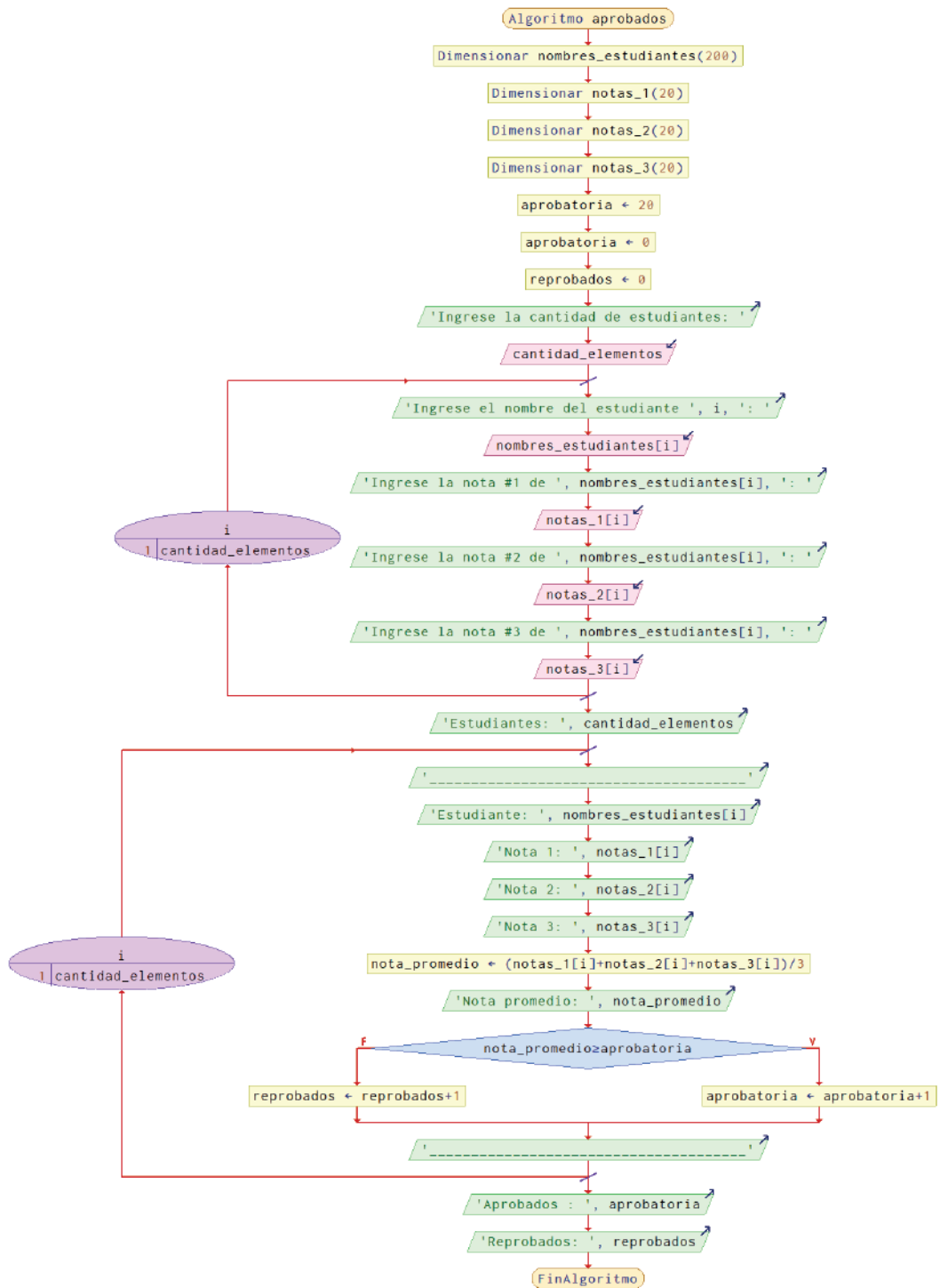
Algoritmo aprobados

```
Dimension nombres_estudiantes[200];
Dimension notas_1[20];
Dimension notas_2[20];
Dimension notas_3[20];
aprobatoria <- 20
aprobatoria <- 0
reprobados <- 0
Escribir "Ingrese la cantidad de estudiantes:";
Leer cantidad_elementos;
Para i<-1 Hasta cantidad_elementos Hacer
    Escribir "Ingrese el nombre del estudiante ";i";"
    Leer nombres_estudiantes[i];
    Escribir "Ingrese la nota #1 de ",nombres_estudiantes[i];"
    Leer notas_1[i];
    Escribir "Ingrese la nota #2 de ",nombres_estudiantes[i];"
```

```
Leer notas_2[i];
Escribir "Ingrese la nota #3 de ",nombres_estudiantes[i]," :";
Leer notas_3[i];
FinPara
Escribir "Estudiantes:", cantidad_elementos;
Para i<-1 Hasta cantidad_elementos Hacer
    Escribir " _____ ";
    Escribir "Estudiante:", nombres_estudiantes[i];
    Escribir "Nota 1:", notas_1[i];
    Escribir "Nota 2:", notas_2[i];
    Escribir "Nota 3:", notas_3[i];
    nota_promedio <- (notas_1[i]+notas_2[i]+notas_3[i])/3;
    Escribir "Nota promedio:", nota_promedio;
    Si nota_promedio >= aprobatoria Entonces
        aprobatoria = aprobatoria + 1
    SiNo
        reprobados = reprobados + 1
    FinSi;
    Escribir " _____ ";
FinPara
Escribir "Aprobados :", aprobatoria;
Escribir "Reprobados:", reprobados;
```

FinAlgoritmo

4.14.2. Diagrama de flujo



4.14.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio14Bloque4 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int totalAlumnos = 5;

        int[] calificacionesExamen1 = new int[totalAlumnos];
        int[] calificacionesExamen2 = new int[totalAlumnos];
        int[] calificacionesExamen3 = new int[totalAlumnos];

        System.out.println("Ingrese las calificaciones de los alumnos para cada examen:");

        for (int i = 0; i < totalAlumnos; i++) {

            System.out.println("Alumno " + (i + 1) + ":");
            System.out.print("Examen 1:");
            calificacionesExamen1[i] = scanner.nextInt();
            System.out.print("Examen 2:");
            calificacionesExamen2[i] = scanner.nextInt();
            System.out.print("Examen 3:");
            calificacionesExamen3[i] = scanner.nextInt();

        }

        int alumnosAprobaronTodos = 0;

        for (int i = 0; i < totalAlumnos; i++) {

            if (calificacionesExamen1[i] > 11 && calificacionesExamen2[i] > 11 && calificacionesExamen3[i] > 11) {
```



```
    alumnosAprobaronTodos++;
    }
}

int alumnosAprobaronAlMenosUno = 0;
for (int i = 0; i < totalAlumnos; i++) {
    if (calificacionesExamen1[i] > 11 || calificacionesExamen2[i] > 11 || calificacionesExamen3[i] > 11) {
        alumnosAprobaronAlMenosUno++;
    }
}

int alumnosAprobaronUltimo = 0;
for (int i = 0; i < totalAlumnos; i++) {
    if (calificacionesExamen3[i] > 11 && calificacionesExamen1[i] <= 11 && calificacionesExamen2[i] <= 11) {
        alumnosAprobaronUltimo++;
    }
}

System.out.println("Número de alumnos que aprobaron todos los exámenes:"
+ alumnosAprobaronTodos);

System.out.println("Número de alumnos que aprobaron al menos un examen:"
+ alumnosAprobaronAlMenosUno);

System.out.println("Número de alumnos que aprobaron únicamente el último
examen:" + alumnosAprobaronUltimo);
}
}
```

4.15.

Realice un programa que solicite al usuario que piense un número entero entre el 1 y el 100. El programa debe generar un número aleatorio en ese mismo rango [1-100], e indicarle al usuario si el número que digito es menor o mayor al número aleatorio, así hasta que lo adivine. Y por último mostrarle el número de intentos que le llevo.

4.15.1. Código Pseudint

Algoritmo B4_Ejercicio_15

Definir numero, numeroAleatorio, intentos Como Entero

numeroAleatorio <- Aleatorio(1, 100)

intentos <- 0

Escribir "Piensa un número entero entre 1 y 100."

Repetir

 Escribir "Adivina el número:"

 Leer numero

 intentos <- intentos + 1

 Si numero < numeroAleatorio Entonces

 Escribir "El número que digitaste es menor al número aleatorio."

 FinSi

 Si numero > numeroAleatorio Entonces

 Escribir "El número que digitaste es mayor al número aleatorio."

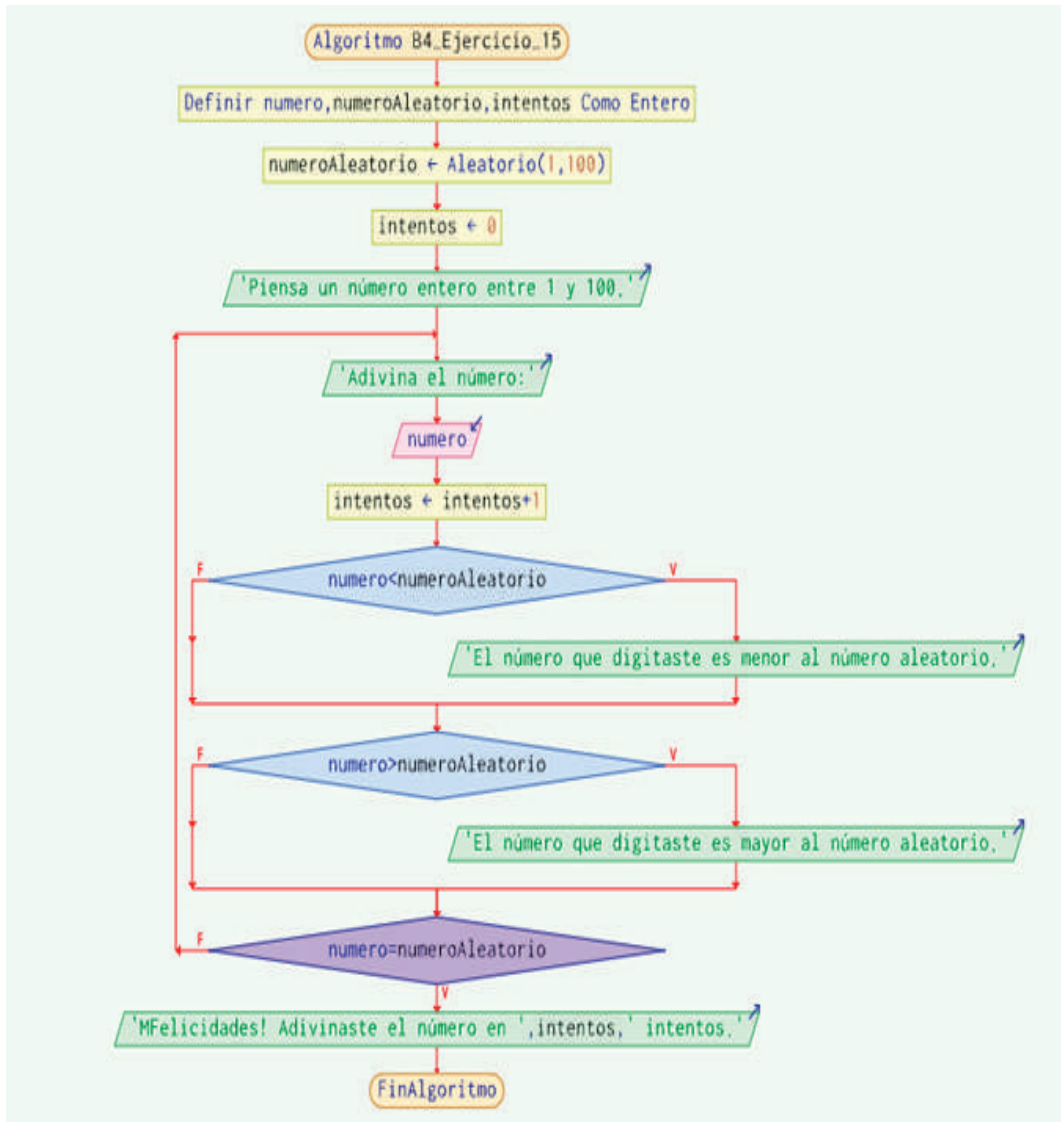
 FinSi

Hasta Que numero = numeroAleatorio

Escribir "¡Felicidades! Adivinaste el número en ", intentos, " intentos."

FinAlgoritmo

4.15.2. Diagrama de flujo



4.15.3. Adaptación a Java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int numero, aleatorio, intentos = 0;
        Scanner entrada = new Scanner(System.in);
        aleatorio = (int) (Math.random() * 100 + 1);
        do{
            System.out.println("Digite un número:");
            numero = entrada.nextInt();
            intentos += 1;
            if (numero != aleatorio){
                if (numero > aleatorio){
                    System.out.println(numero + " es mayor que el generado, intente de nuevo");
                } else{
                    System.out.println(numero + " es menor que el generado, intente de nuevo");
                }
            }
        } while (numero != aleatorio);
        System.out.println("FELICITACIONES, le tomó " + intentos + " intentos");
    }
}
```

4.16.

Realice un programa que calcule la descomposición en factores primos de un número entero. Por ejemplo: $20 = 2 \cdot 2 \cdot 5$.

4.16.1. Código Pseint

Algoritmo Ejercicio 081

```
Definir num,candidato Como Entero
num <- 0
Escribir 'Dime un número'
Leer num
candidato <- 1
Mientras num>1 Hacer
    candidato <- PrimoSiguiente(candidato)
    Si num MOD candidato=0 Entonces
        Escribir candidato
        num <- num/candidato
        candidato <- 1
    FinSi
FinMientras
```

FinAlgoritmo

Funcion resultado <- Primo (num)

Definir divisores,i Como Entero

divisores <- 0

i <- 0

Definir resultado Como Logico

resultado <- Falso

Para i<-num Hasta 1 Con Paso -1 Hacer

Si num MOD i=0 Entonces

divisores <- divisores+1

FinSi

Si divisores=2 Entonces

resultado <- Verdadero

FinSi

FinPara

FinFuncion

Funcion resultado <- PrimoSiguiente (num)

Definir resultado Como Entero

Definir encontrado Como Logico

encontrado <- Falso

resultado <- 0

Mientras encontrado=Falso Hacer

num <- num+1

encontrado <- Primo(num)

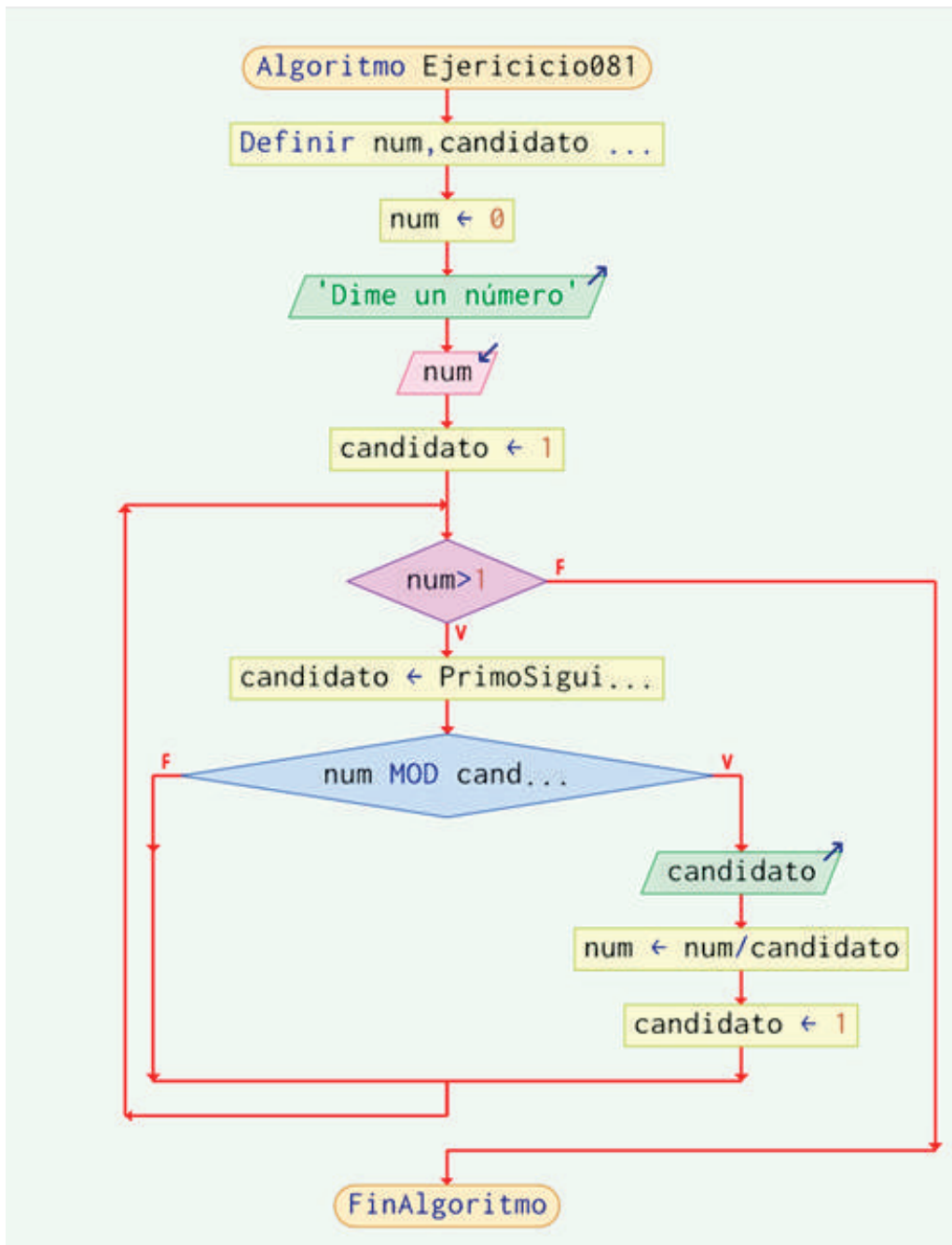
FinMientras

resultado <- num

FinFuncion

4.16.2. Diagrama de flujo

NOTA: El Pseudocódigo, tiene varios diagramas de flujo, pero donde hace la función es en este, ya que es en el diagrama de flujo donde se lo hace correr no en el pseudocódigo.



4.16.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio16Bloque4 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese un número entero:");

        int numero = scanner.nextInt();

        System.out.print("Descomposición en factores primos de " + numero + ":");

        descomponerEnFactoresPrimos(numero);

        System.out.println();

    }

    public static void descomponerEnFactoresPrimos(int numero) {

        int factor = 2;

        while (numero > 1) {

            if (numero % factor == 0) {

                // El factor es un divisor del número

                System.out.print(factor + " ");

                numero /= factor;

            } else {

                factor++;

            }

        }

    }

}
```


Bloque 05: Arreglos o vectores

Escribe un programa que defina un vector de números y calcule la suma de sus elementos.

5.1.1. Código Pseint

Algoritmo B5_Ejercicio_1

Definir suma, i, vector, n Como Entero

Escribir "El programa define un vector de números y calcula la suma de sus elementos."

Escribir ""

Escribir "Ingresar tamaño del vector:"

Leer n

Dimension vector[n]

Para i=1 Hasta n Con Paso 1 Hacer

 Escribir "Ingresar el elemento " i

 Leer vector[i]

Fin Para

suma=0

Para i=1 Hasta n Con Paso 1 Hacer

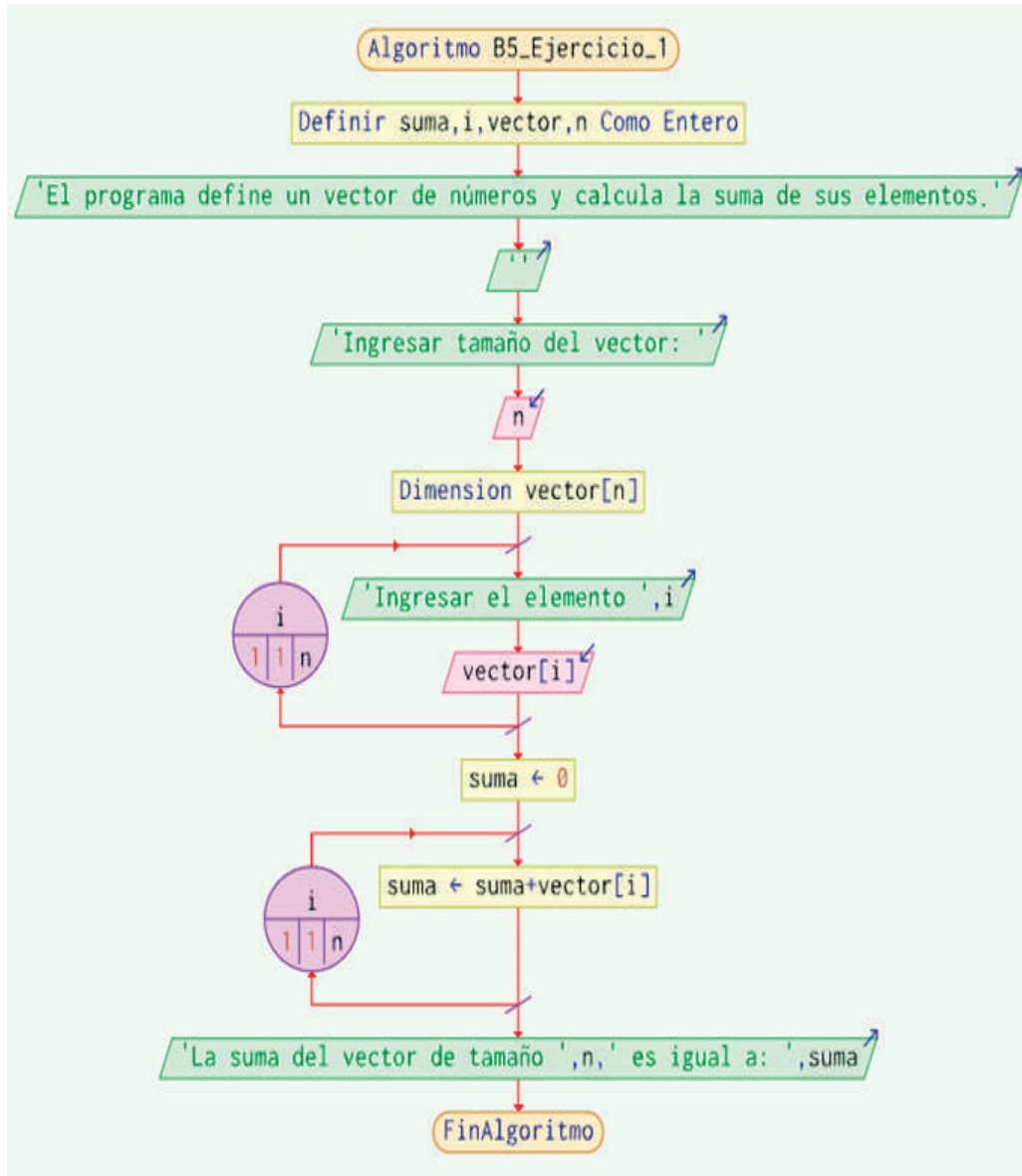
 suma=suma+vector[i]

Fin Para

Escribir "La suma del vector de tamaño " n, " es igual a: " suma

FinAlgoritmo

5.1.2. Diagrama de flujo



5.1.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio1Bloque5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el tamaño del vector:");
        int tamaño = scanner.nextInt();
        int[] vector = new int[tamaño];
        System.out.println("Ingrese los elementos del vector:");
        for (int i = 0; i < tamaño; i++) {
            System.out.print("Elemento " + (i + 1) + ":");
            vector[i] = scanner.nextInt();
        }
        int suma = 0;
        for (int i = 0; i < tamaño; i++) {
            suma += vector[i];
        }
        System.out.println("La suma de los elementos del vector es: " + suma);
    }
}
```

5.2.

Escribe un programa que defina un vector de números y calcule la multiplicación acumulada de sus elementos.

5.2.1. Código Pseint

Algoritmo vector

Definir vectorA,vectorB,vectorC,x Como Entero

Dimension vectorA[5], vectorB[5],vectorC[5]

para x = 1 Hasta 5 Con Paso 1 Hacer

 Escribir "Ingresa el numero ", x

 leer vectorA(x)

FinPara

para x = 1 Hasta 5 Con Paso 1 Hacer

 Escribir "Ingresa el numero ", x

 leer vectorB(x)

FinPara

para x = 1 Hasta 5 Con Paso 1 Hacer

 vectorC(x) = vectorA(x) * vectorB(x)

FinPara

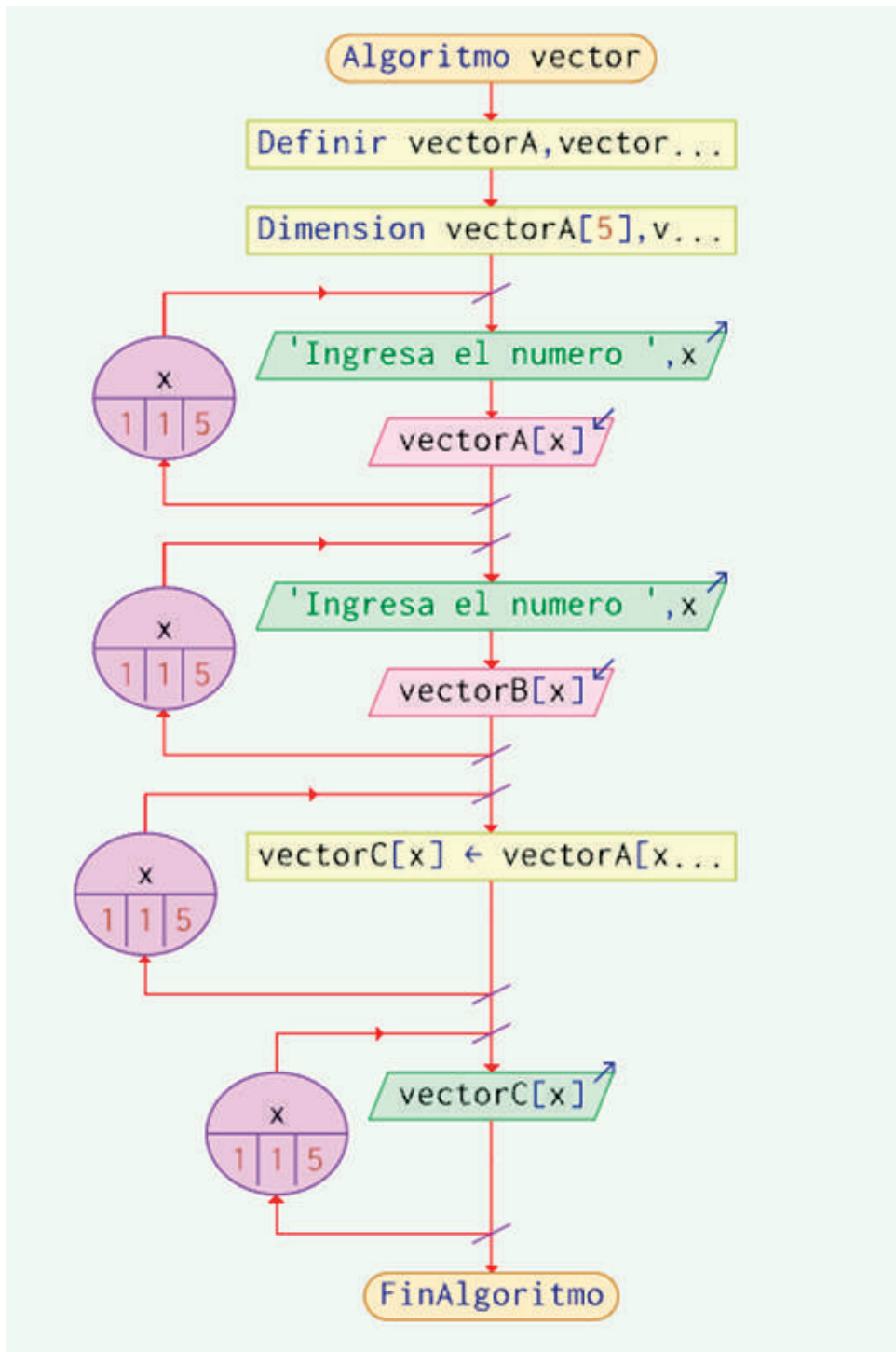
para x = 1 Hasta 5 Con Paso 1 Hacer

 Escribir vectorC(x)

FinPara

FinAlgoritmo

5.2.2. Diagrama de flujo



5.2.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio2Bloue5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el tamaño del vector:");
        int tamaño = scanner.nextInt();
        int[] vector = new int[tamaño];
        System.out.println("Ingrese los elementos del vector:");
        for (int i = 0; i < tamaño; i++) {
            System.out.print("Elemento " + (i + 1) + " :");
            vector[i] = scanner.nextInt();
        }
        int multiplicacion = 1;
        for (int i = 0; i < tamaño; i++) {
            multiplicacion *= vector[i];
        }
        System.out.println("La multiplicación acumulada de los elementos del vector es:
        " + multiplicacion);
    }
}
```

5.3.

Escribe un programa que lea de la entrada estándar un vector de números y muestre en la salida estándar los números del vector con sus índices asociados.

5.3.1. Código Pseint

Algoritmo B5_Ejercicio_3

Definir i, vector, n Como Entero

Escribir “El programa lee desde la entrada estándar un vector de números y muestra en la salida estándar los números del vector con sus índices asociados”

Escribir “”

Escribir “Ingresar tamaño del vector:”

Leer n

Dimension vector[n]

Para i=1 Hasta n Con Paso 1 Hacer

 Escribir “Ingresar el elemento “ i

 Leer vector[i]

Fin Para

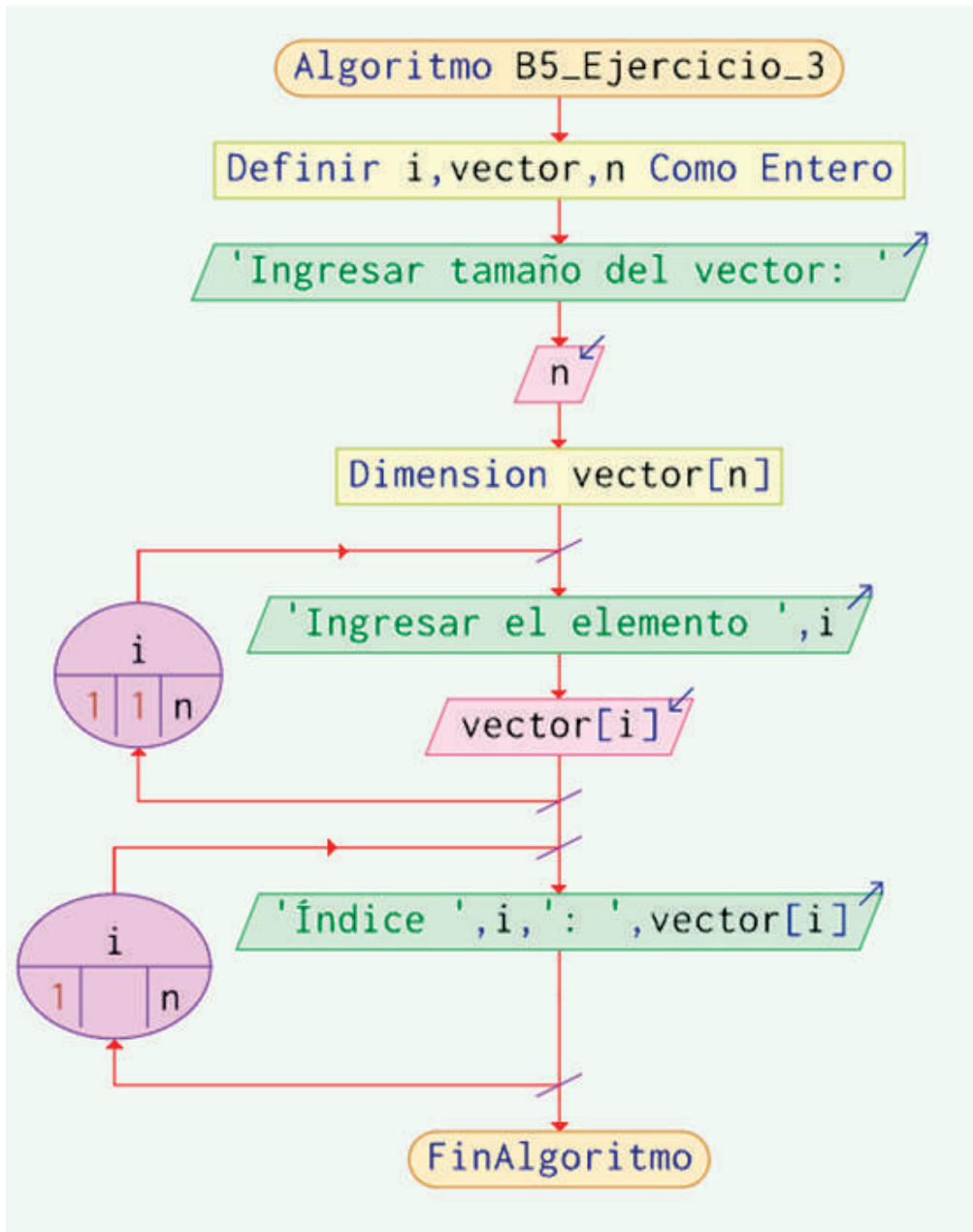
Para i = 1 Hasta n Hacer

 Escribir “Índice “ i, “:“, vector[i]

FinPara

FinAlgoritmo

5.3.2. Diagrama de flujo



5.3.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio3Bloque5 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el tamaño del vector:");

        int tamaño = scanner.nextInt();

        int[] vector = new int[tamaño];

        System.out.println("Ingrese los elementos del vector:");

        for (int i = 0; i < tamaño; i++) {

            System.out.print("Elemento " + (i + 1) + ":");

            vector[i] = scanner.nextInt();

        }

        System.out.println("Números del vector con sus índices asociados:");

        for (int i = 0; i < tamaño; i++) {

            System.out.println("Índice " + i + ":" + vector[i]);

        }

    }

}
```

54.

Escribe un programa que defina un vector de números y muestre en la salida estándar el vector en orden inverso—del último al primer elemento.

54.1. Código Pseint**Algoritmo INVERSA**

Definir x,vector1,vector2 Como Entero

Dimensión vector1[10],vector2[10]

para x = 1 Hasta 10 Con Paso 1 Hacer

 Escribir “Ingresa el numero”,x

 leer vector1(x)

FinPara

para x = 10 Hasta 1 Con Paso -1 Hacer

 vector2(x) = vector1(x)

FinPara

Escribir ““

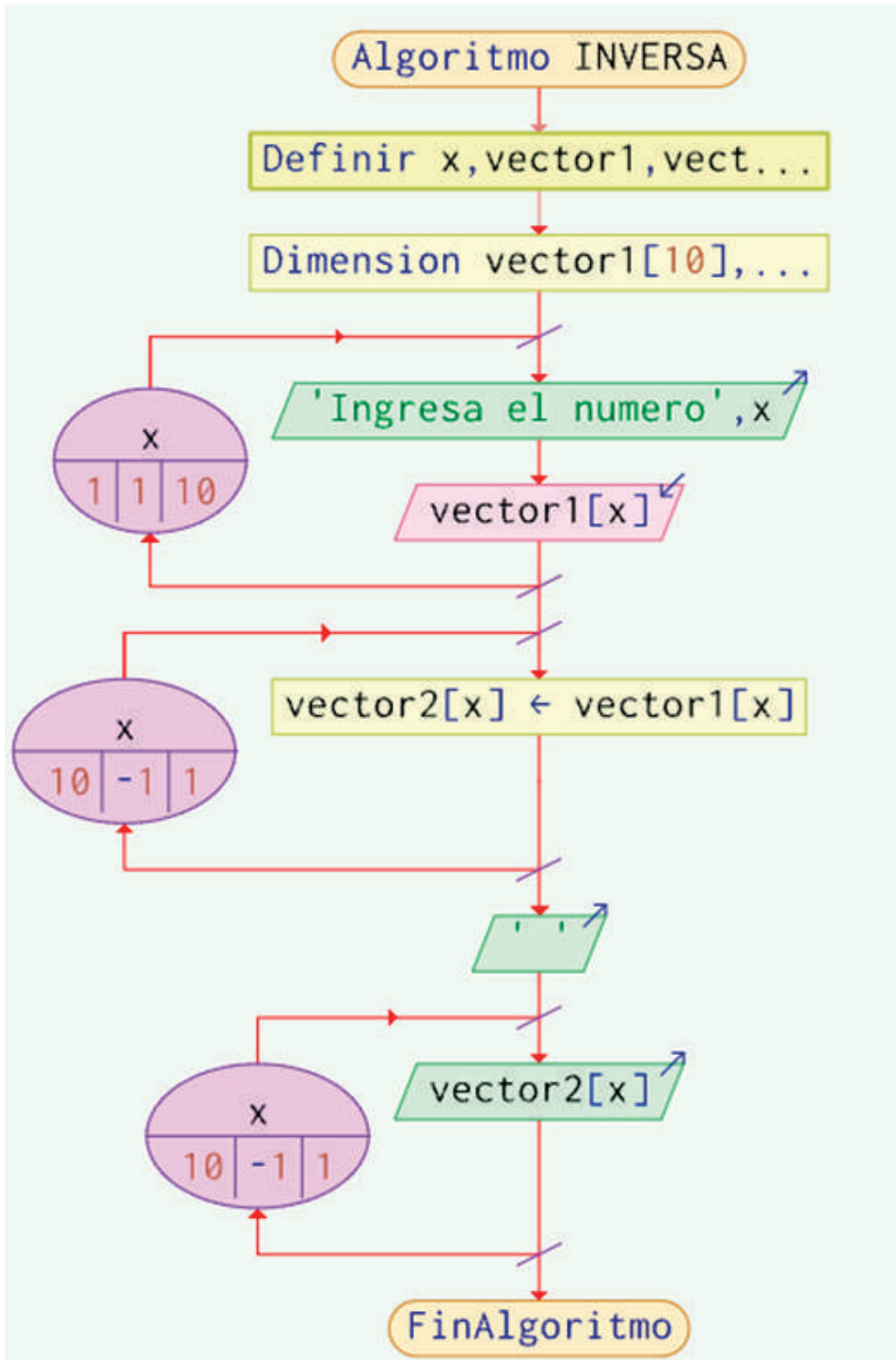
para x = 10 Hasta 1 Con Paso -1 Hacer

 escribir vector2(x)

FinPara

FinAlgoritmo

54.2. Diagrama de Flujo



54.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio4Bloque5{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el tamaño del vector:");

        int tamaño = scanner.nextInt();

        int[] vector = new int[tamaño];

        System.out.println("Ingrese los elementos del vector:");

        for (int i = 0; i < tamaño; i++) {

            System.out.print("Elemento " + (i + 1) + " :");

            vector[i] = scanner.nextInt();

        }

        System.out.println("Vector en orden inverso:");

        for (int i = tamaño - 1; i >= 0; i--) {

            System.out.println(vector[i]);

        }

    }

}
```

5.5.

Desarrolle un programa que lea de la entrada estándar un vector de enteros y determine el mayor elemento del vector.

5.5.1. Código Pseint

Algoritmo B5_Ejercicio_5

Definir i, n, nro, mayor, cant Como Entero

Escribir "El programa lee desde la entrada estándar un vector de enteros y determina el mayor elemento del vector."

Escribir ""

Escribir "Ingresar tamaño del vector:"

Leer cant

Dimension nro[cant]

Para i=1 Hasta cant Con Paso 1 Hacer

 Escribir "Ingresar el elemento:", i

 Leer n

 nro[i] = n

Fin Para

Para i=1 Hasta cant Con Paso 1 Hacer

 Si (i==1) Entonces

 mayor=nro[i]

 SiNo

 Si (nro[i]>mayor) Entonces

 mayor=nro[i]

 SiNo

 Fin Si

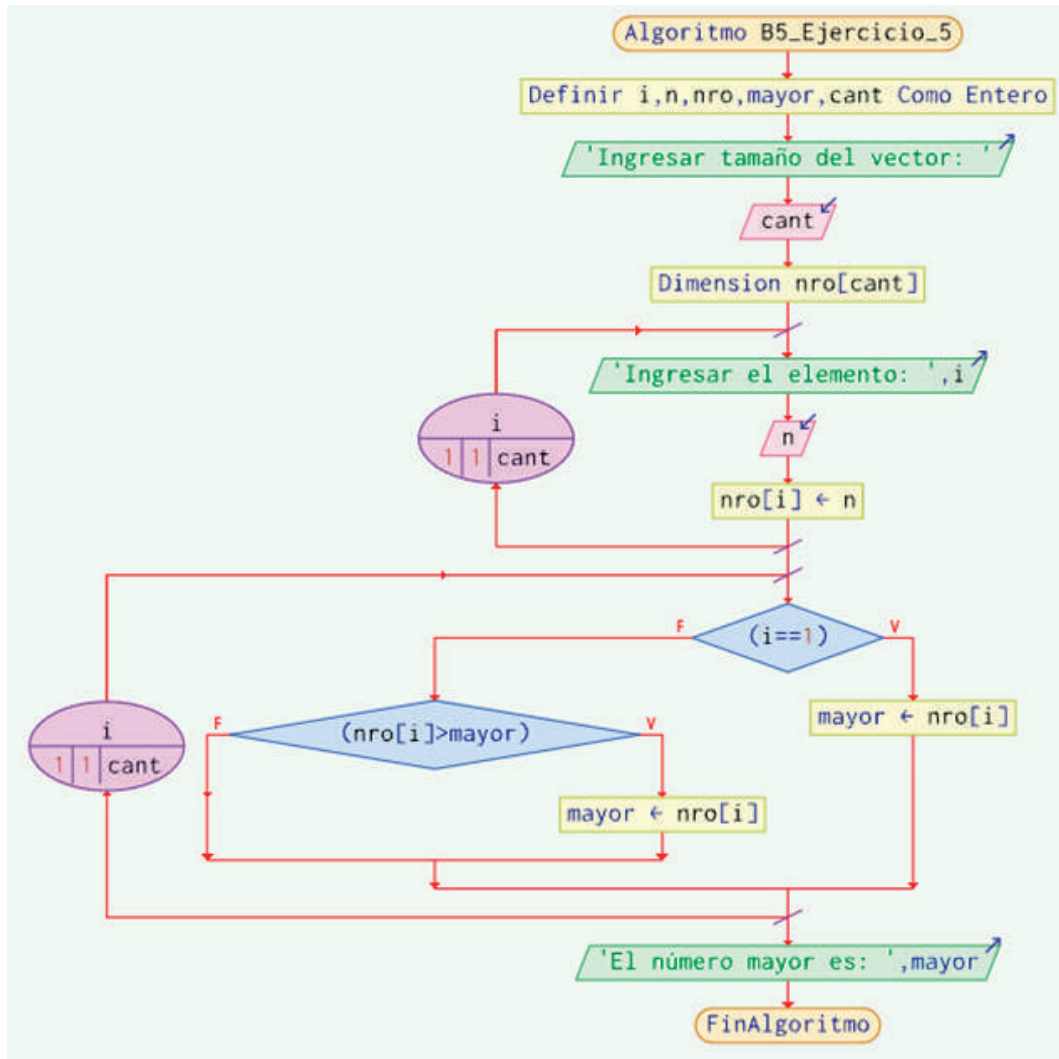
Fin Para

Fin Para

Escribir "El número mayor es:", mayor

FinAlgoritmo

5.5.2. Diagrama de flujo



5.5.3. Adaptación a Java

```
import java.util.Scanner;
public class Ejercicio5Bloque5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el tamaño del vector:");
        int tamaño = scanner.nextInt();
        int[] vector = new int[tamaño];
        System.out.println("Ingrese los elementos del vector:");
        for (int i = 0; i < tamaño; i++) {
            System.out.print("Elemento " + (i + 1) + ":");
            vector[i] = scanner.nextInt();
        }
        int mayor = vector[0];
        for (int i = 1; i < tamaño; i++) {
            if (vector[i] > mayor) {
                mayor = vector[i];
            }
        }
        System.out.println("El mayor elemento del vector es: " + mayor);
    }
}
```

5.6.

Escribe un programa que defina un vector de números y calcule si existe algún número en el vector cuyo valor equivale a la suma del resto de números del vector.

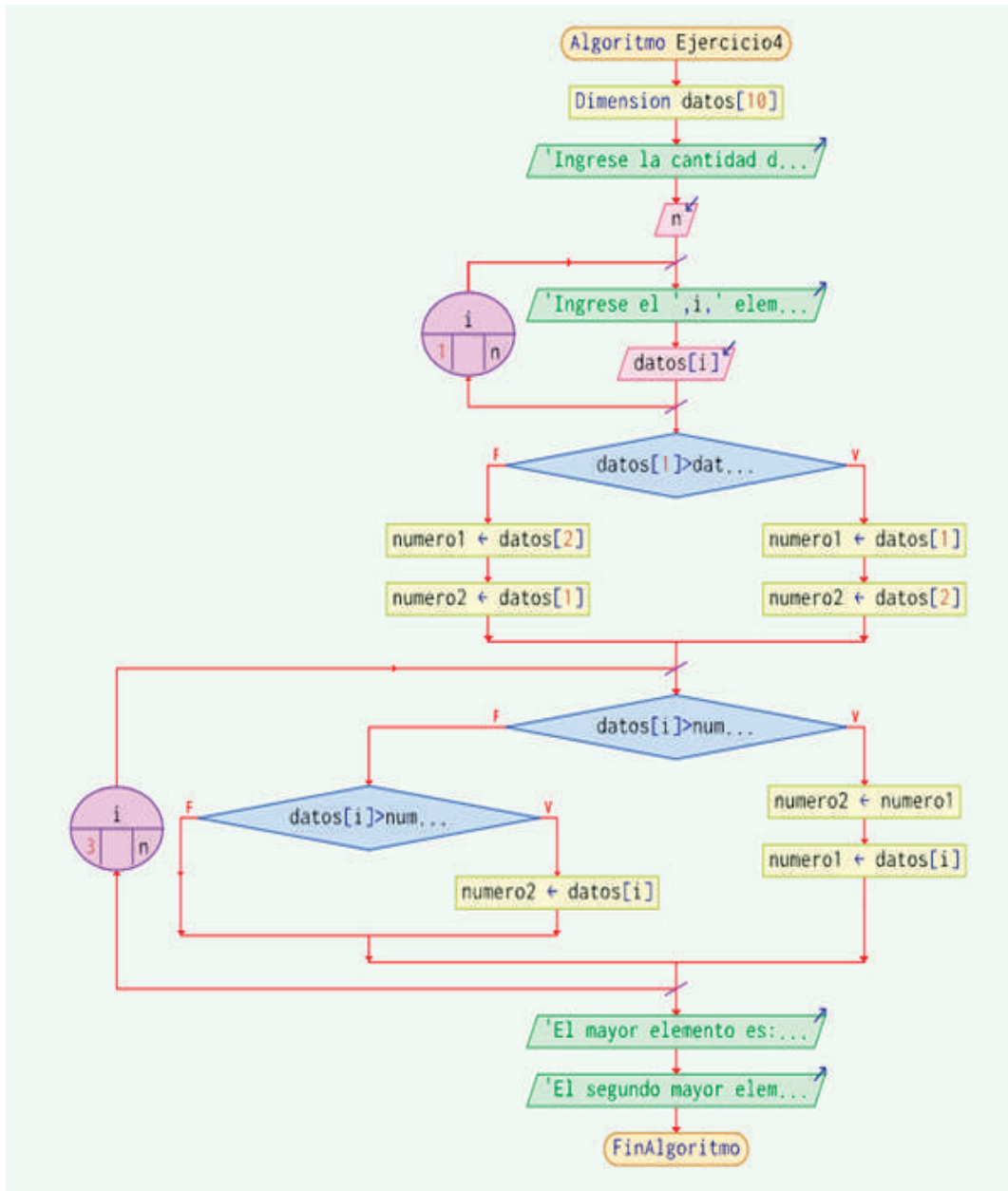
5.6.1. Código Pseint

Algoritmo Ejercicio4

```
Dimension datos[10];  
  
Escribir "Ingrese la cantidad de elementos:";  
Leer n;  
Para i<-1 Hasta n Hacer  
  Escribir "Ingrese el " i " elemento";  
  Leer datos[i];  
FinPara  
  
Si datos[1]>datos[2] Entonces  
  numero1<-datos[1];  
  numero2<-datos[2];  
Sino  
  
  numero1<-datos[2];  
  
  numero2<-datos[1];  
  
FinSi  
  
Para i<-3 Hasta n Hacer  
  Si datos[i]>numero1 Entonces  
    numero2<-numero1;  
    numero1<-datos[i];  
  
  Sino  
  
    Si datos[i]>numero2 Entonces  
      numero2<-datos[i];  
    FinSi  
  FinSi  
FinPara  
Escribir "El mayor elemento es:",numero1;  
  
Escribir "El segundo mayor elemento es:",numero2;
```

FinAlgoritmo

5.6.2. Diagrama de flujo



5.6.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio6Bloque5 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el tamaño del vector:");

        int tamaño = scanner.nextInt();
        int[] vector = new int[tamaño];

        System.out.println("Ingrese los elementos del vector:");

        for (int i = 0; i < tamaño; i++) {
            System.out.print("Elemento " + (i + 1) + ":");
            vector[i] = scanner.nextInt();
        }

        boolean encontrado = false;
        for (int i = 0; i < tamaño; i++) {
            int sumaResto = 0;
            for (int j = 0; j < tamaño; j++) {
                if (j != i) {
                    sumaResto += vector[j];
                }
            }
            if (vector[i] == sumaResto) {
                encontrado = true;
                break;
            }
        }
        if (encontrado) {
            System.out.println("Existe al menos un número cuyo valor es igual a la suma
del resto de números del vector.");
        } else {

            System.out.println("No existe ningún número cuyo valor sea igual a la suma
del resto de números del vector.");
        }
    }
}
```

5.7.

Realiza un programa que defina dos vectores de caracteres y después almacene el contenido de ambos vectores en un nuevo vector, situando en primer lugar los elementos del primer vector seguido por los elementos del segundo vector. Muestre el contenido del nuevo vector en la salida estándar.

5.7.1. Código Pseint

Algoritmo B5_Ejercicio_7

```
Definir longitud1, longitud2, i, j Como Entero
Definir vector1, vector2, vectorConcatenado Como Caracter
Escribir "Ingrese la longitud del primer vector:"
Leer longitud1
Dimension vector1 [longitud1]

Escribir "Ingrese el longitud del segundo vector:"
Leer longitud2
Dimension vector2 [longitud2]

Para i = 1 Hasta longitud1 Hacer
    Escribir "Ingrese el elemento ", i, " del primer vector:"
    Leer vector1 [i]
FinPara

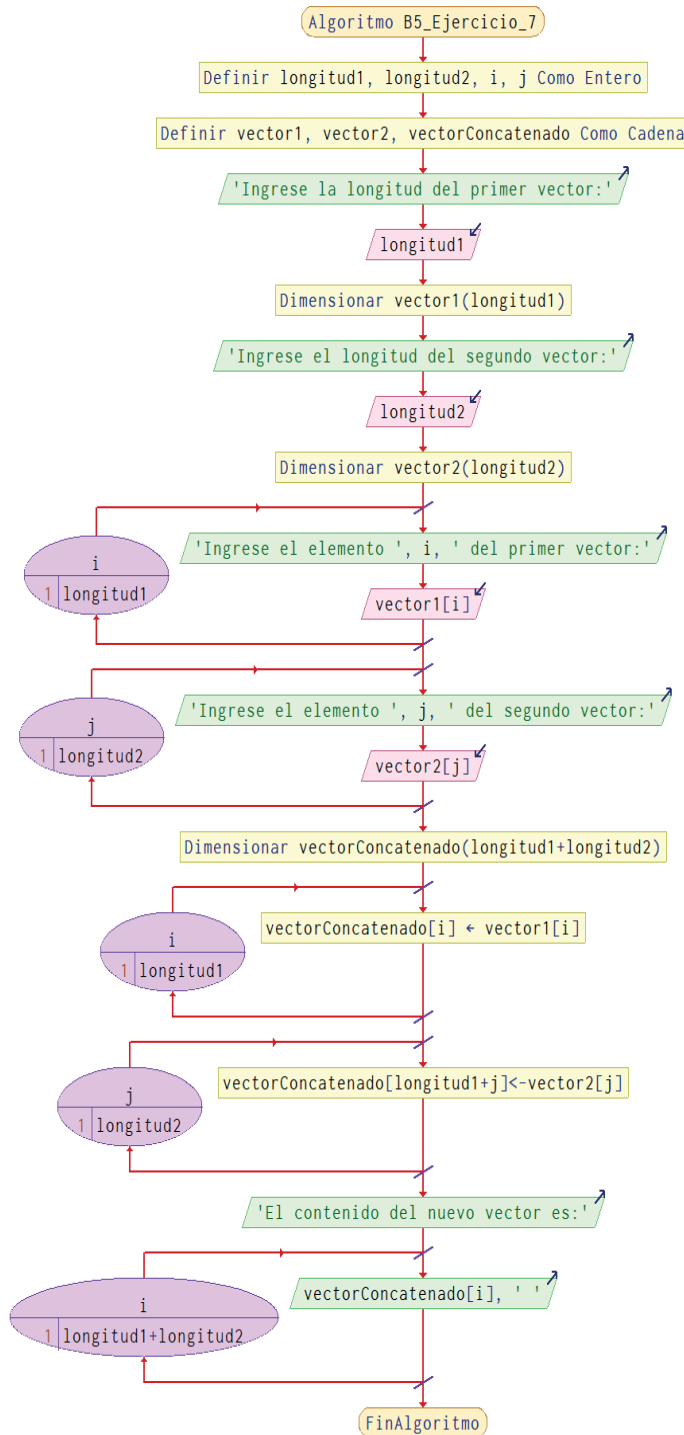
Para j = 1 Hasta longitud2 Hacer
    Escribir "Ingrese el elemento ", j, " del segundo vector:"
    Leer vector2 [j]
FinPara

Dimension vectorConcatenado [longitud1 + longitud2]
Para i = 1 Hasta longitud1 Hacer
    vectorConcatenado [i] = vector1 [i]
FinPara
Para j = 1 Hasta longitud2 Hacer
    vectorConcatenado [longitud1 + j] = vector2 [j]
FinPara

Escribir "El contenido del nuevo vector es:"
Para i = 1 Hasta longitud1 + longitud2 Hacer
    Escribir SinSaltar vectorConcatenado [i], " "
FinPara
```

FinAlgoritmo

5.7.2. Diagrama de flujo



5.7.3. Adaptación a Java

```
public class Ejercicio7Bloque5 {  
    public static void main(String[] args) {  
        char[] vector1 = {'H', 'o', 'l', 'a'};  
        char[] vector2 = {'M', 'u', 'n', 'd', 'o'};  
        int tamañoNuevoVector = vector1.length + vector2.length;  
        char[] nuevoVector = new char[tamañoNuevoVector];  
        for (int i = 0; i < vector1.length; i++) {  
            nuevoVector[i] = vector1[i];  
        }  
        for (int i = 0; i < vector2.length; i++) {  
            nuevoVector[vector1.length + i] = vector2[i];  
        }  
        System.out.println("Contenido del nuevo vector:");  
        for (char elemento : nuevoVector) {  
            System.out.print(elemento + " ");  
        }  
    }  
}
```

5.8.

Hacer un programa que lea 5 números en un arreglo, los copie a otro arreglo, multiplicado por 2 y muestre el segundo arreglo.

5.8.1. Código Pseint

Algoritmo multiplicacion

Definir a,b,c,d,e,mul Como Entero

Escribir "Ingresa los cinco números"

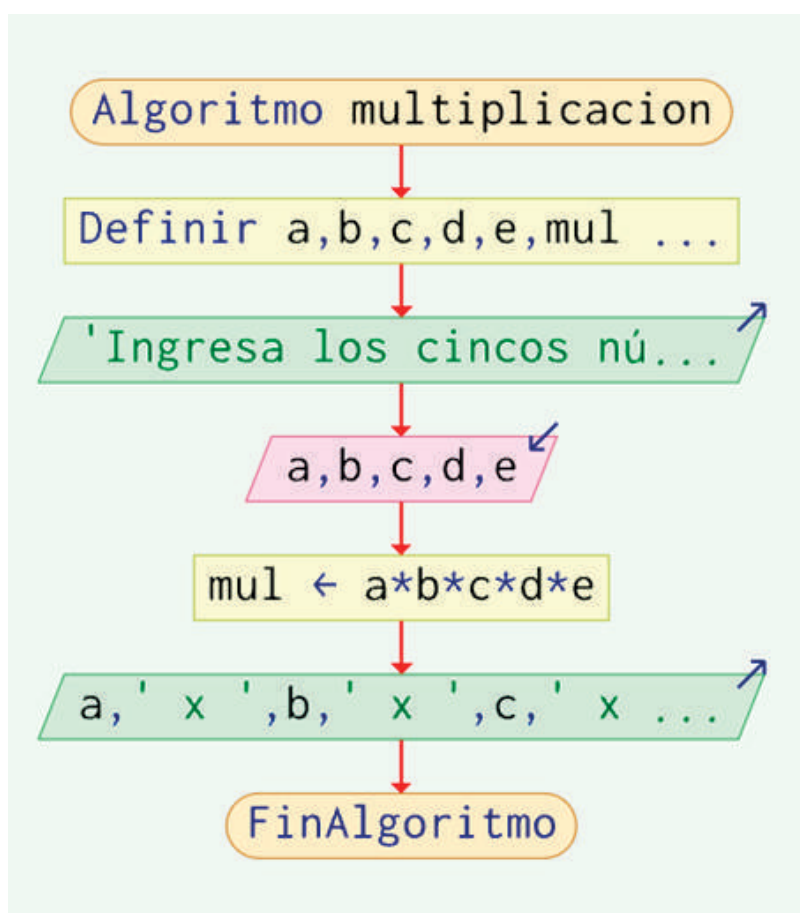
leer a,b,c,d,e

$mul = a * b * c * d * e$

Escribir a, " x ", b, " x ", c, " x ", d, " x ", e, " = ", $a*b*c*d*e$

FinAlgoritmo

5.8.2. Diagrama de flujo



5.8.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio8Bloque5 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int[] arreglo1 = new int[5];

        System.out.println("Ingrese 5 números:");

        for (int i = 0; i < 5; i++) {

            System.out.print("Número " + (i + 1) + ":");

            arreglo1[i] = scanner.nextInt();

        }

        int[] arreglo2 = new int[5];

        for (int i = 0; i < 5; i++) {

            arreglo2[i] = arreglo1[i] * 2;

        }

        System.out.println("Contenido del segundo arreglo (multiplicado por 2):");

        for (int i = 0; i < 5; i++) {

            System.out.println(arreglo2[i]);

        }

    }

}
```

Bloque 06: Matrices o tablas

6.1.

Hacer un programa para rellenar una matriz pidiendo al usuario el número de filas y columnas, Posteriormente mostrar la matriz en pantalla.

6.1.1. Código Pseint

Algoritmo B6_Ejercicio_I

Definir numFilas, numColumnas, i, j Como Entero

Definir matriz Como Entero

Escribir “El programa rellena una matriz pidiendo al usuario el número de filas y columnas. Posteriormente muestra la matriz en pantalla.”

Escribir “”

Escribir “Ingrese el número de filas de la matriz:”

Leer numFilas

Escribir “Ingrese el número de columnas de la matriz:”

Leer numColumnas

Dimension matriz[numFilas, numColumnas]

Para i <- 1 Hasta numFilas Hacer

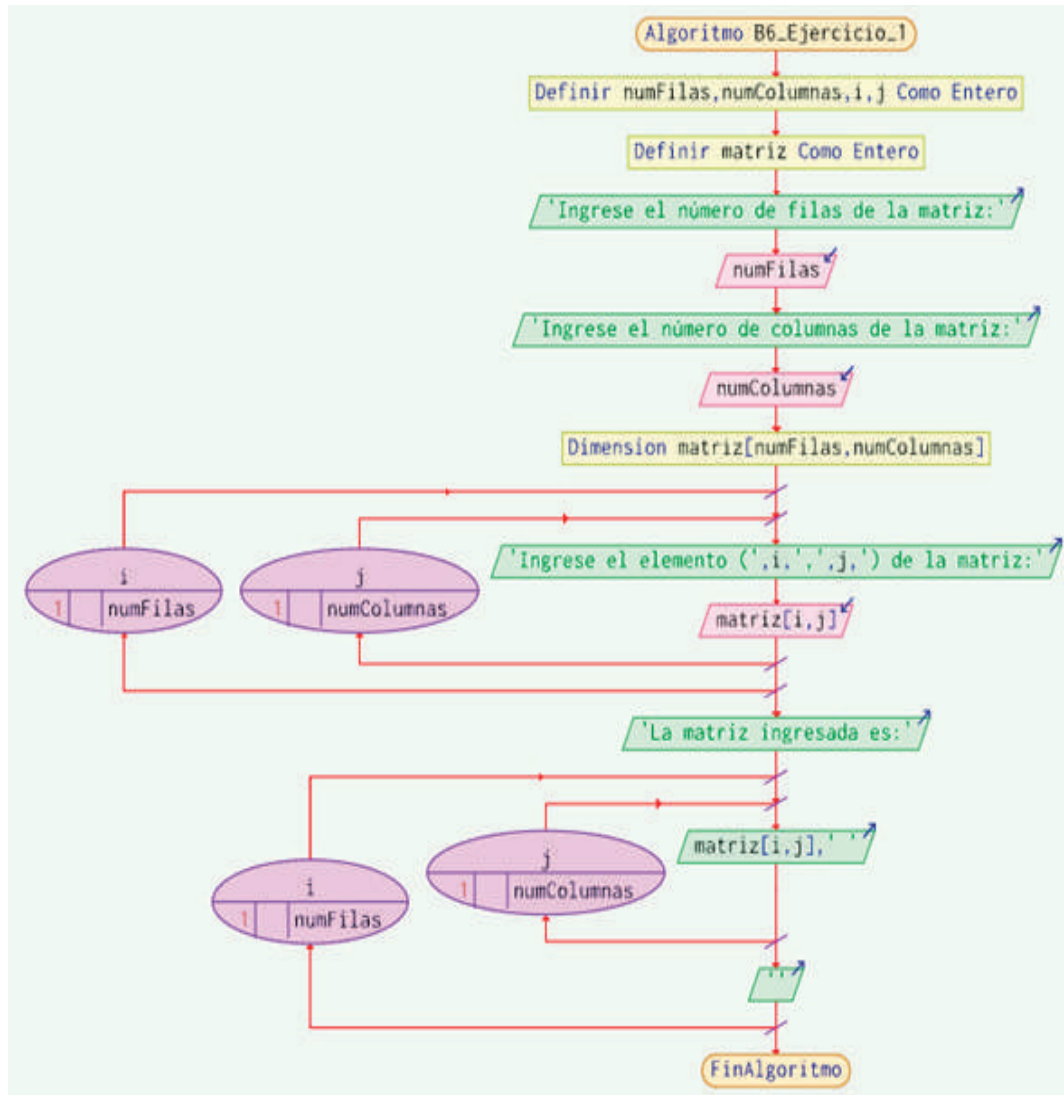
Para j <- 1 Hasta numColumnas Hacer

Escribir “Ingrese el elemento (“i”, “j”) de la matriz:”

Leer matriz[i, j]


```
FinPara
FinPara
Escribir "La matriz ingresada es:"
Para i <- 1 Hasta numFilas Hacer
    Para j <- 1 Hasta numColumnas Hacer
        Escribir Sin Saltar matriz[i, j], ""
    FinPara
    Escribir ""
FinPara
FinAlgoritmo
```

6.1.2. Diagrama de flujo



6.1.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio8Bloque5 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int[] arreglo1 = new int[5];

        System.out.println("Ingrese 5 números:");

        for (int i = 0; i < 5; i++) {

            System.out.print("Número " + (i + 1) + ":");

            arreglo1[i] = scanner.nextInt();

        }

        int[] arreglo2 = new int[5];

        for (int i = 0; i < 5; i++) {

            arreglo2[i] = arreglo1[i] * 2;

        }

        System.out.println("Contenido del segundo arreglo (multiplicado por 2):");

        for (int i = 0; i < 5; i++) {

            System.out.println(arreglo2[i]);

        }

    }

}
```

6.2.

Realiza un programa que defina una matriz de 3x3 y escriba un ciclo para que muestre la diagonal principal de la matriz.

6.2.1. Código Pseint

Algoritmo ejercicio4 (nombres, FILAS)

Para i<-1 Hasta FILAS Con Paso 1 Hacer

 escribir “La longitud del nombre “,nombres[i, 1],” es de “,nombres[i, 2]

Fin Para

Fin SubProceso

Proceso Ejercicio6

 Definir longitud_nombres como entero

 Definir nombres como cadena

 Definir FILAS como entero

 definir num_nombre como cadena

 escribir “Introduce las filas del array multidimensional”

 leer FILAS

 Dimension nombres[FILAS, 2]

 Para i<-1 Hasta FILAS Con Paso 1 Hacer

 escribir “Introduce un nombre en la fila “,i

 leer nombres[i, 1]

 num_nombre<-convertirtexto(Longitud(nombres[i, 1]))

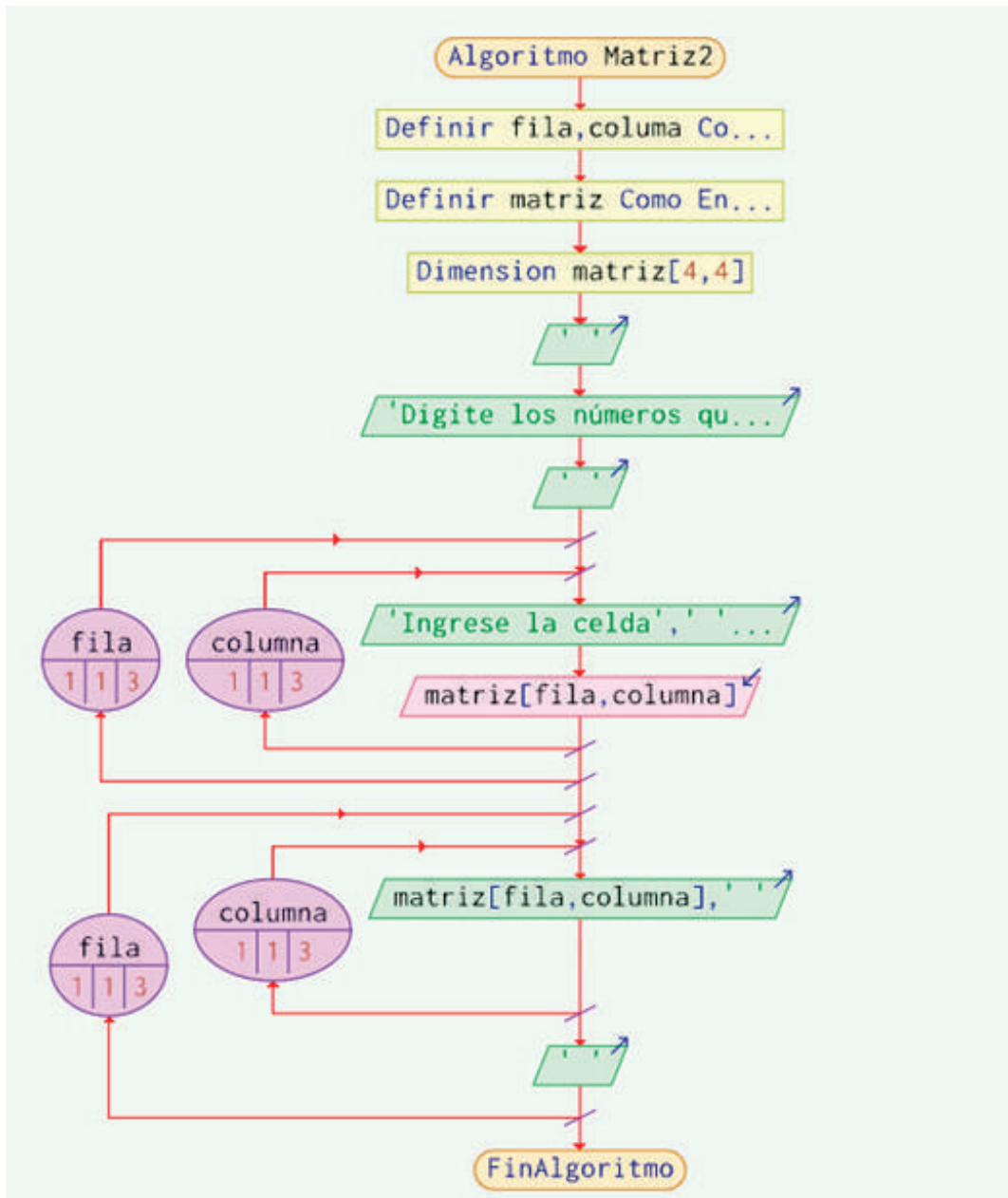
 nombres[i, 2]<-num_nombre

 Fin Para

 mostrarArrays(nombres, FILAS)

FinAlgoritmo

6.2.2. Diagrama de flujo



6.2.3. Adaptación a Java

```
public class Ejercicio2Bloque6 {  
    public static void main(String[] args) {  
        int[][] matriz = {{1, 2, 3},  
                           {4, 5, 6},  
                           {7, 8, 9}};  
  
        System.out.println("Diagonal principal:");  
        for (int i = 0; i < matriz.length; i++) {  
            System.out.println(matriz[i][i]);  
        }  
    }  
}
```

6.3.

Hacer una matriz de tipo entera de $2 * 2$, llenarla de números y luego copiar todo su contenido hacia otra matriz.

6.3.1. Código Pseint

Algoritmo B6_Ejercicio_3

Dimension matrizA[2, 2], matrizB[2, 2]

Definir i, j Como Entero

Escribir “El programa hacer una matriz de tipo entera de 2 * 2, luego se llena de números y luego copia todo su contenido hacia otra matriz.”

Escribir “”

Escribir “Ingrese los valores de la matriz A:”

Para i = 1 Hasta 2 Hacer

Para j = 1 Hasta 2 Hacer

Escribir “Ingrese el elemento “, i, “”, j

Leer matrizA[i, j]

FinPara

FinPara

Para i = 1 Hasta 2 Hacer

Para j = 1 Hasta 2 Hacer

matrizB[i, j] <- matrizA[i, j]

FinPara

FinPara

Escribir “La matriz B es:”

Para i = 1 Hasta 2 Hacer

Para j = 1 Hasta 2 Hacer

Escribir Sin Saltar matrizB[i, j], “ “

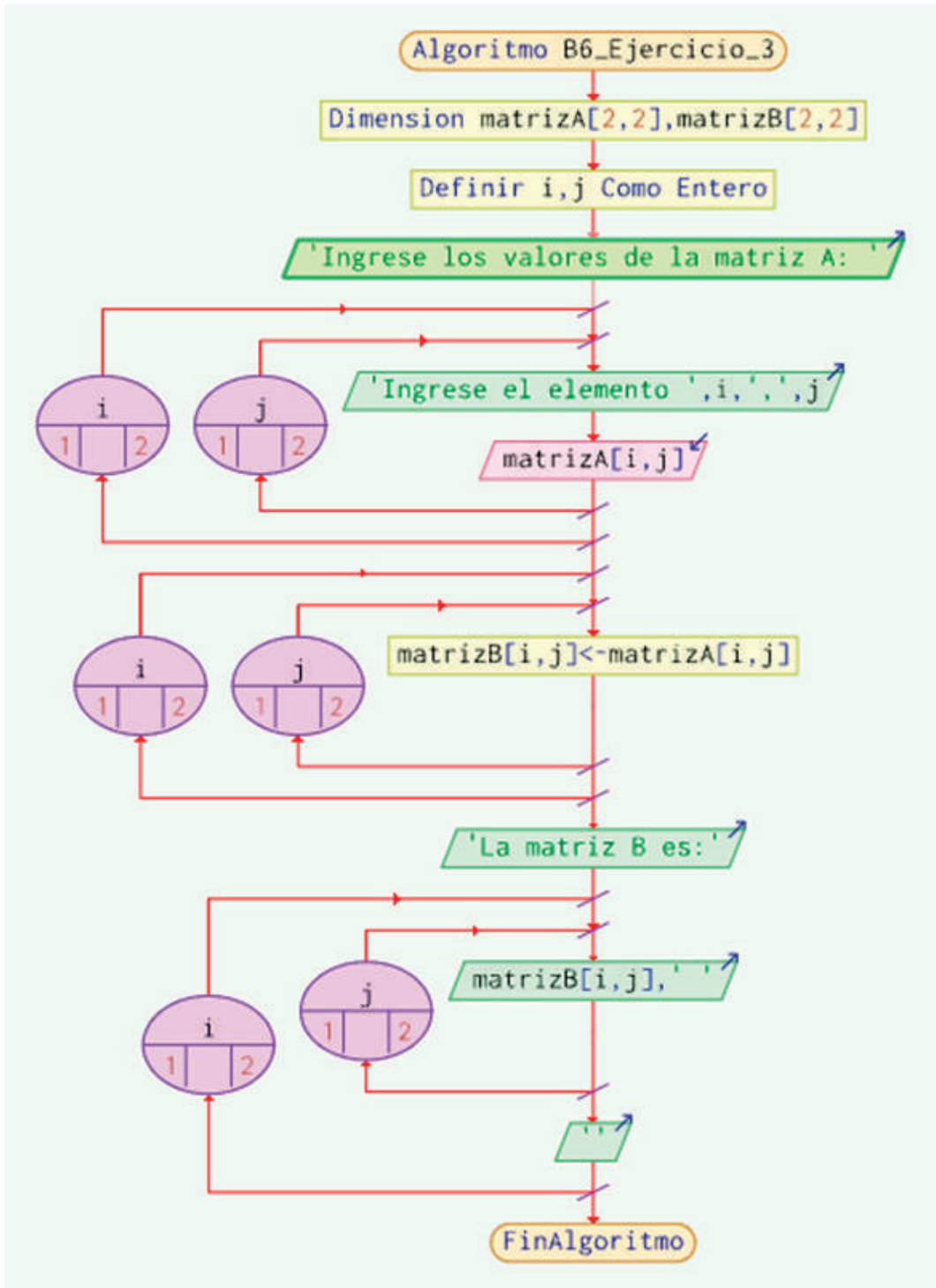
FinPara

Escribir “”

FinPara

FinAlgoritmo

6.3.2. Diagrama de flujo



6.3.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio3Bloque6 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[][] matriz1 = new int[2][2];

        System.out.println("Ingrese los valores de la matriz:");

        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                System.out.print("Valor en la posición (" + i + ", " + j + "):");
                matriz1[i][j] = scanner.nextInt();
            }
        }

        int[][] matriz2 = new int[2][2];
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                matriz2[i][j] = matriz1[i][j];
            }
        }

        System.out.println("Matriz copiada:");

        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                System.out.print(matriz2[i][j] + " ");
            }

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

64.

Hacer una matriz de tipo entera de $2 * 2$, llenarla de números y luego copiar todo su contenido hacia otra matriz.

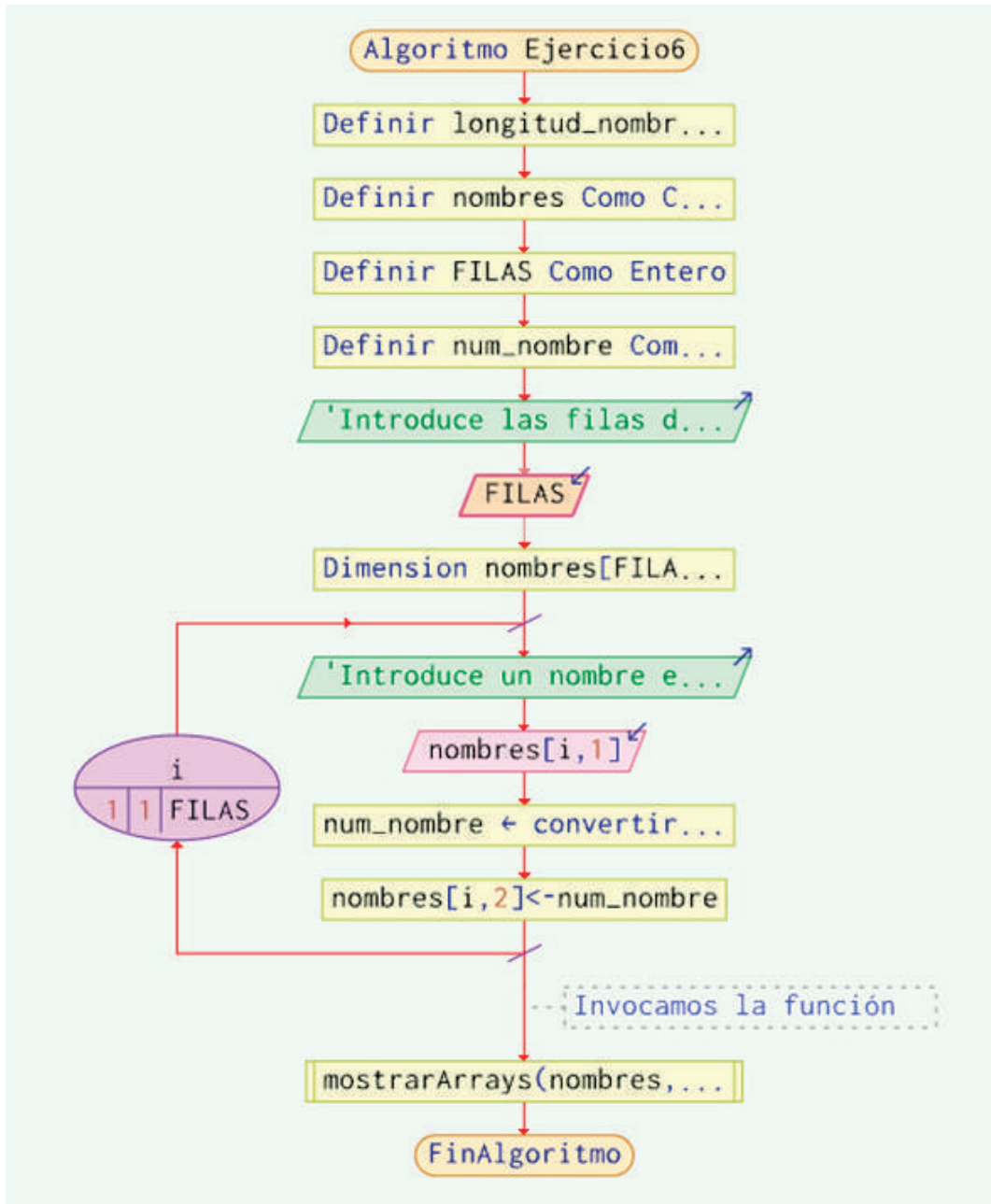
64.1. Código Pseint

SubProceso mostrarArrays (nombres, FILAS)

```
Para i<-1 Hasta FILAS Con Paso 1 Hacer
    escribir "La longitud del nombre ",nombres[i,1]," es de ",nombres[i,2]
Fin Para
Fin SubProceso
Proceso Ejercicio6
    Definir longitud_nombres como entero
    Definir nombres como cadena
    Definir FILAS como entero
    definir num_nombre como cadena
    escribir "Introduce las filas del array multidimensional"
    leer FILAS
    Dimension nombres[FILAS,2]
    Para i<-1 Hasta FILAS Con Paso 1 Hacer
        escribir "Introduce un nombre en la fila ",i
        leer nombres[i,1]
        num_nombre<-convertirtexto(Longitud(nombres[i,1]))
        nombres[i,2]<-num_nombre
    Fin Para
    //Invocamos la función
    mostrarArrays( nombres, FILAS)
```

FinProceso

64.2. Diagrama de flujo



64.3. Adaptación a Java

```
import java.util.Random;

import java.util.Scanner;

public class Ejercicio4Bloque6 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de filas:");

        int filas = scanner.nextInt();

        System.out.print("Ingrese el número de columnas:");
        int columnas = scanner.nextInt();

        int[][] matrizOriginal = new int[filas][columnas];
        int[][] matrizCopia = new int[filas][columnas];

        Random random = new Random();

        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                matrizOriginal[i][j] = random.nextInt(100);
            }
        }
        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                matrizCopia[i][j] = matrizOriginal[i][j];
            }
        }
        System.out.println("Matriz copiada:");

        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                System.out.print(matrizCopia[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

6.5.

Realice un programa que lea una matriz de 3x3 y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

6.5.1. Código Pseint

Algoritmo B6_Ejercicio_5

```
Definir matriz, traspuesta Como Entero
Dimension matriz[3, 3]
Dimension traspuesta[3, 3]

Definir i, j Como Entero

Escribir "Ingrese los elementos de la matriz (3x3):"

Para i <- 1 Hasta 3 Hacer
  Para j <- 1 Hasta 3 Hacer
    Escribir "Ingrese el elemento (" , i, ",", j, "):"
    Leer matriz[i, j]
  FinPara
FinPara

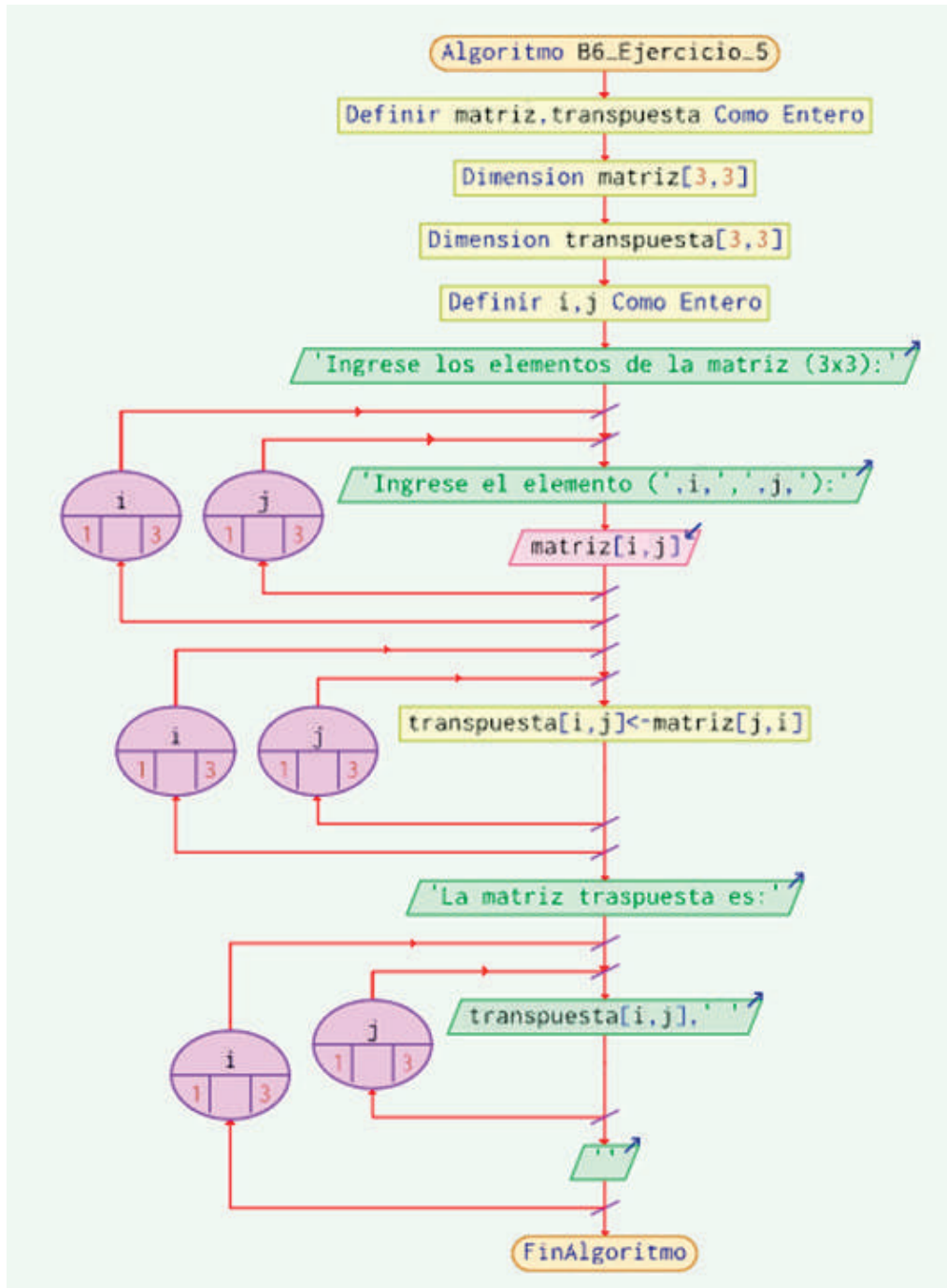
Para i <- 1 Hasta 3 Hacer
  Para j <- 1 Hasta 3 Hacer
    traspuesta[i, j] <- matriz[j, i]
  FinPara
FinPara

Escribir "La matriz traspuesta es:"

Para i <- 1 Hasta 3 Hacer
  Para j <- 1 Hasta 3 Hacer
    Escribir Sin Saltar traspuesta[i, j], " "
  FinPara
Escribir ""
FinPara
```

FinAlgoritmo

6.5.2. Diagrama de flujo



6.5.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio5Bloque6 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int[][] matriz = new int[3][3];

        System.out.println("Ingrese los valores de la matriz:");

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                System.out.print("Valor en la posición (" + i + ", " + j + "):");

                matriz[i][j] = scanner.nextInt();

            }

        }

        int[][] traspuesta = new int[3][3];

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                traspuesta[i][j] = matriz[j][i];

            }

        }

        System.out.println("Matriz traspuesta:");

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                System.out.print(traspuesta[i][j] + " ");

            }

            System.out.println();

        }

    }

}
```

6.6.

Realice un programa que calcule la suma de dos matrices cuadradas de 3x3.

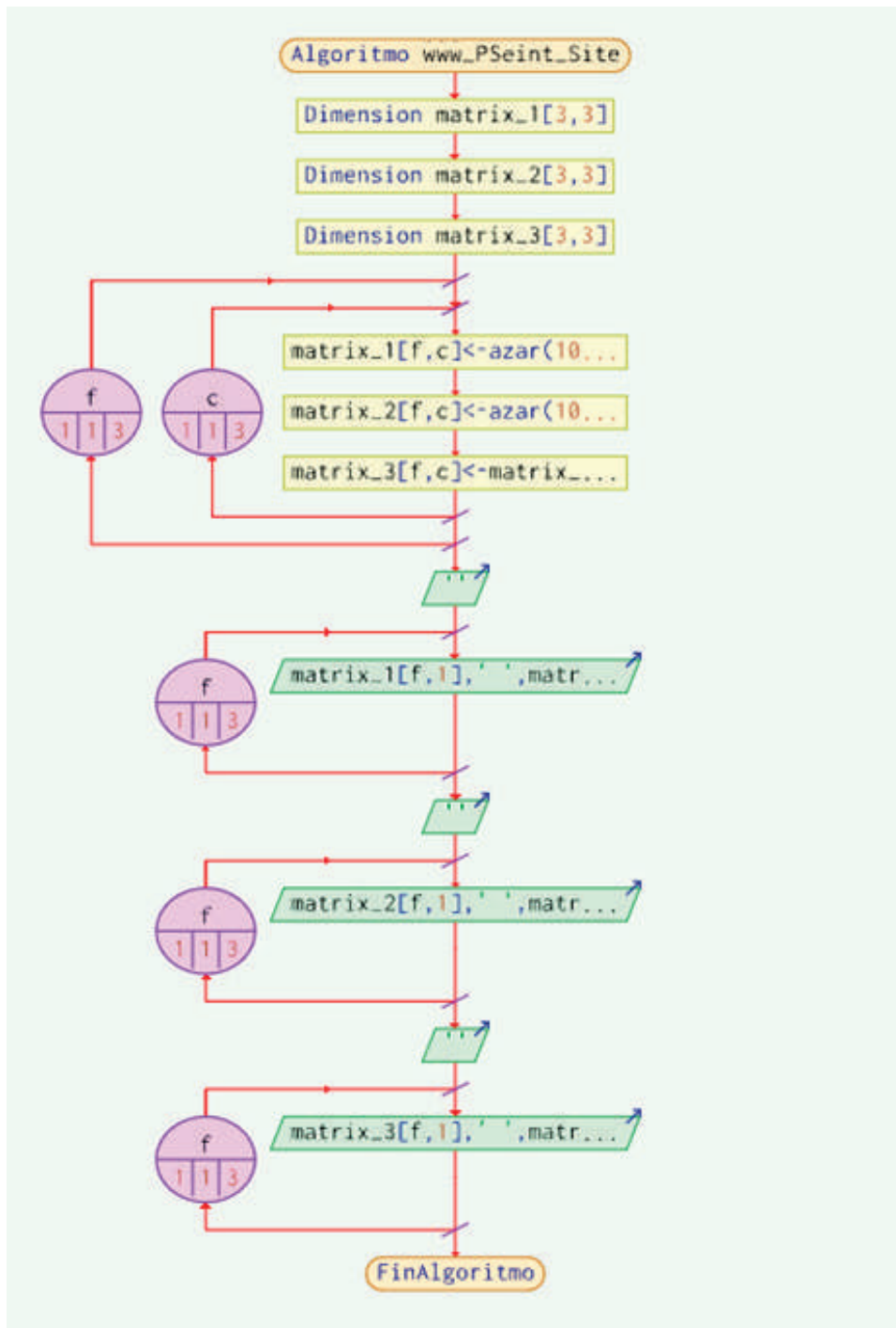
6.6.1. Código Pseint

Algoritmo `www_PSeint_Site`

```
Dimension matrix_1[3,3]
Dimension matrix_2[3,3]
Dimension matrix_3[3,3]
Para f = 1 hasta 3 con paso 1 hacer
    Para c = 1 hasta 3 con paso 1 hacer
        matrix_1[f,c] = azar(100) + 1
        matrix_2[f,c] = azar(100) + 1
        matrix_3[f,c] = matrix_1[f,c] + matrix_2[f,c]
    FinPara
FinPara
Escribir ""
Para f = 1 hasta 3 con paso 1 hacer
    Escribir matrix_1[f,1] , "" , matrix_1[f,2] , "" , matrix_1[f,3]
FinPara
Escribir ""
Para f = 1 hasta 3 con paso 1 hacer
    Escribir matrix_2[f,1] , "" , matrix_2[f,2] , "" , matrix_2[f,3]
FinPara
Escribir ""
Para f = 1 hasta 3 con paso 1 hacer
    Escribir matrix_3[f,1] , "" , matrix_3[f,2] , "" , matrix_3[f,3]
FinPara
```

FinAlgoritmo

6.6.2. Diagrama de flujo



6.6.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio6Bloque6 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[][] matriz1 = new int[3][3];

        System.out.println("Ingrese los valores de la primera matriz:");

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {

                System.out.print("Valor en la posición (" + i + ", " + j + "):");
                matriz1[i][j] = scanner.nextInt();
            }
        }

        int[][] matriz2 = new int[3][3];
        System.out.println("Ingrese los valores de la segunda matriz:");

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("Valor en la posición (" + i + ", " + j + "):");
                matriz2[i][j] = scanner.nextInt();
            }
        }

        int[][] resultado = new int[3][3];
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                resultado[i][j] = matriz1[i][j] + matriz2[i][j];
            }
        }

        System.out.println("Matriz resultado:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(resultado[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

6.7.

esarrollar un programa que determine si una matriz es simétrica o no. Una matriz es simétrica si es cuadrada y si es igual a su matriz transpuesta.

6.7.1. Código Pseint

Algoritmo B6_Ejercicio_7

Definir n, matriz, transpuesta Como Entero

Definir esSimetrica Como Logico

Escribir "Ingrese el tamaño de la matriz cuadrada (n x n):"

Leer n

Dimension matriz[n, n]

Dimension transpuesta[n, n]

Escribir "Ingrese los elementos de la matriz:"

Para i <- 1 Hasta n Hacer

Para j <- 1 Hasta n Hacer

Escribir "Ingrese el elemento (" , i, " , " , j, "):"

Leer matriz[i, j]

FinPara

FinPara

Para i <- 1 Hasta n Hacer

Para j <- 1 Hasta n Hacer

transpuesta[i, j] <- matriz[j, i]

FinPara

FinPara

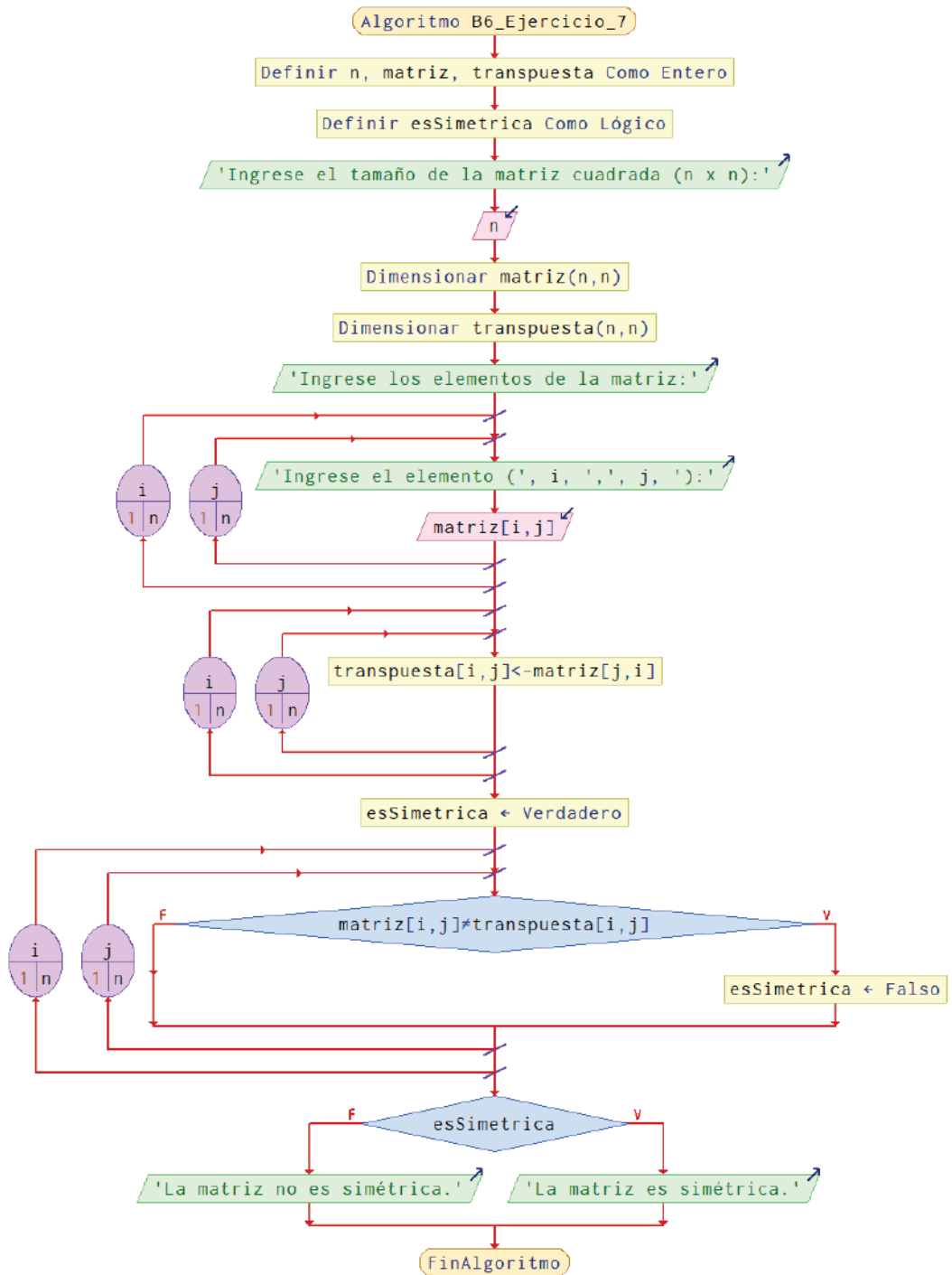
esSimetrica <- Verdadero

Para i <- 1 Hasta n Hacer

Para j <- 1 Hasta n Hacer

```
Si matriz[i, j] <> transpuesta[i, j] Entonces
    esSimetrica <- Falso
FinSi
FinPara
FinPara
Si esSimetrica Entonces
    Escribir "La matriz es simétrica."
Sino
    Escribir "La matriz no es simétrica."
FinSi
FinAlgoritmo
```

6.7.2. Diagrama de flujo



6.7.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio7Blquie6 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de filas y columnas de la matriz:");
        int n = scanner.nextInt();

        int[][] matriz = new int[n][n];

        System.out.println("Ingrese los valores de la matriz:");

        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n; j++) {

                System.out.print("Valor en la posición (" + i + ", " + j + "):");
                matriz[i][j] = scanner.nextInt();
            }
        }

        boolean esSimetrica = true;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {

                if (matriz[i][j] != matriz[j][i]) {

                    esSimetrica = false;
                    break;
                }
            }
            if (!esSimetrica) {
                break;
            }
        }
        if (esSimetrica) {
            System.out.println("La matriz es simétrica.");
        } else {
            System.out.println("La matriz no es simétrica.");
        }
    }
}
```

6.8.

Realice un programa que calcule el producto de dos matrices cuadradas de 3x3.

6.8.1. Código Pseint

Algoritmo producto

```
Dimension Matriz_A[2,2]
Dimension Matriz_B[2,2]
Dimension Matriz_Producto[2,2]
Definir Matriz_A,Matriz_B,Matriz_Producto Como Entero
Definir i,j Como Entero
Definir n,m Como Entero
Escribir 'Llenar la Matriz A es:'
Para i<-0 Hasta l Hacer
    Para j<-0 Hasta l Hacer
        Escribir 'Ingrese valor para la fila 'i', en columna 'j,':'
        Leer Matriz_A[i,j]
    FinPara
FinPara
Escribir 'La Matriz A es:'
Para i<-0 Hasta l Hacer
    Para j<-0 Hasta l Hacer
        Escribir Matriz_A[i,j], ' Sin Saltar'
    FinPara
FinPara
Escribir 'Llenar la Matriz B es:'
Para m<-0 Hasta l Hacer
    Para n<-0 Hasta l Hacer
```

Escribir 'Ingrese valor para la fila 'm,' en columna 'n,''

Leer Matriz_B[m,n]

FinPara

FinPara

Escribir 'La Matriz B es:'

Para m<-0 Hasta l Hacer

Para n<-0 Hasta l Hacer

Escribir Matriz_B[m,n],' ' Sin Saltar

FinPara

Escribir ' '

FinPara

Matriz_Producto[0,0]<-Matriz_A[0,0]*Matriz_B[0,0]+Matriz_A[0,1]*Matriz_B[1,0]

Matriz_Producto[0,1]<-Matriz_A[0,1]*Matriz_B[0,1]+Matriz_A[0,1]*Matriz_B[1,1]

Matriz_Producto[1,0]<-Matriz_A[1,0]*Matriz_B[0,0]+Matriz_A[1,1]*Matriz_B[1,0]

Matriz_Producto[1,1]<-Matriz_A[1,0]*Matriz_B[0,1]+Matriz_A[1,1]*Matriz_B[1,1]

Escribir 'La Matriz Producto es:'

Para m<-0 Hasta l Hacer

Para n<-0 Hasta l Hacer

Escribir Matriz_Producto[m,n],' ' Sin Saltar

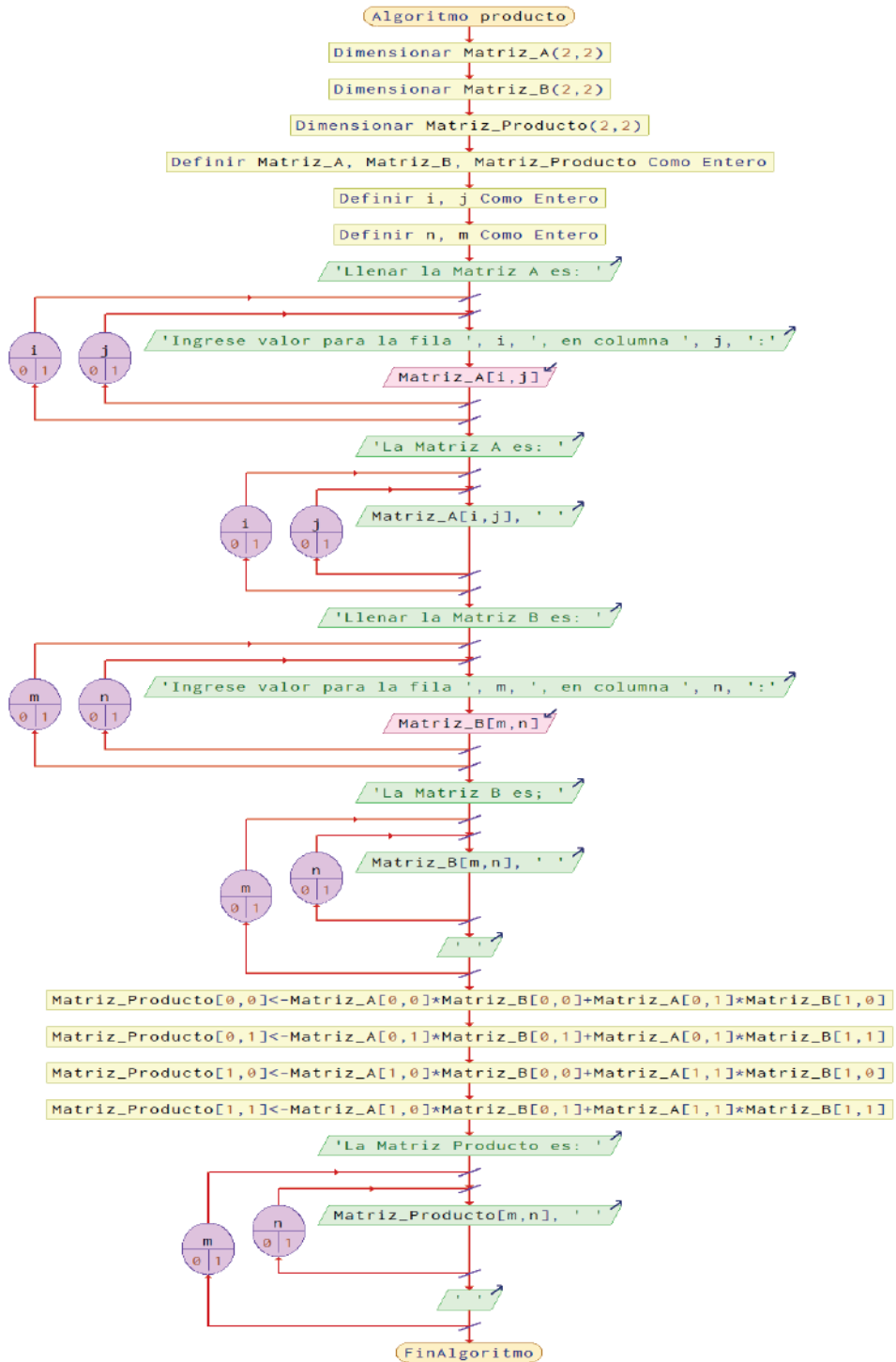
FinPara

Escribir ' '

FinPara

FinAlgoritmo

6.8.2. Diagrama de flujo



6.8.3. Adaptación a Java

```
import java.util.Scanner;

public class Ejercicio7Blquie6 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de filas y columnas de la matriz:");

        int n = scanner.nextInt();
        int[][] matriz = new int[n][n];

        System.out.println("Ingrese los valores de la matriz:");

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {

                System.out.print("Valor en la posición (" + i + ", " + j + "):");
                matriz[i][j] = scanner.nextInt();
            }
        }

        boolean esSimetrica = true;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {

                if (matriz[i][j] != matriz[j][i]) {
                    esSimetrica = false;

                    break;
                }
            }
            if (!esSimetrica) {
                break;
            }
        }
        if (esSimetrica) {
            System.out.println("La matriz es simétrica.");
        } else {

            System.out.println("La matriz no es simétrica.");
        }
    }
}
```

Bloque 07: Cadenas

7.1.

Hacer un programa que pida al usuario que digite una cadena de caracteres, luego verificar la longitud de la cadena, y si ésta supera a 10 caracteres mostrarla en pantalla, caso contrario no mostrarla.

7.1.1. Código Pseint

Algoritmo B7_Ejercicio_1

Escribir “El programa pide al usuario que digite una cadena de caracteres, luego verifica la longitud de la cadena, y si ésta supera los 10 caracteres lo muestra en pantalla, caso contrario no lo muestra.”

Escribir “”

Escribir “Ingrese una cadena de caracteres:”

Leer cadena

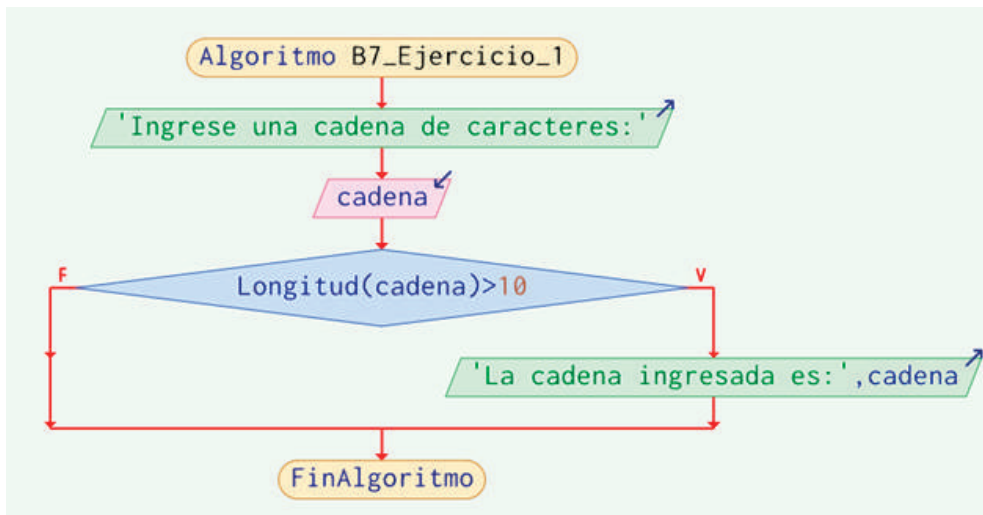
Si Longitud(cadena) > 10 Entonces

 Escribir “La cadena ingresada es:”, cadena

FinSi

FinAlgoritmo

7.1.2. Diagrama de flujo



7.1.3. Adaptación a Java

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese una cadena de caracteres:");

        String cadena = sc.nextLine();

        if (cadena.length() > 10) {

            System.out.println(cadena);

        } else {

            System.out.println("La cadena no supera los 10 caracteres.");

        }

        sc.close();

    }

}
```

7.2.

Pedir al usuario una cadena de caracteres, almacenarla en un arreglo y copiar todo su contenido hacia otro arreglo de caracteres.

7.2.1. Código Pseint

Algoritmo cadena

Definir palabra Como Entero

Escribir "Digite una palabra"

leer pala

size=Longitud(pala)

Escribir " El # de caracteres de la palabra ",pala," es:",size

cade= Subcadena(pala,1,4)

Escribir cade

Dimension letras[size]

Para i<-1 Hasta size Con Paso 1 Hacer

letras[i] = Subcadena(pala,i,i)

Fin Para

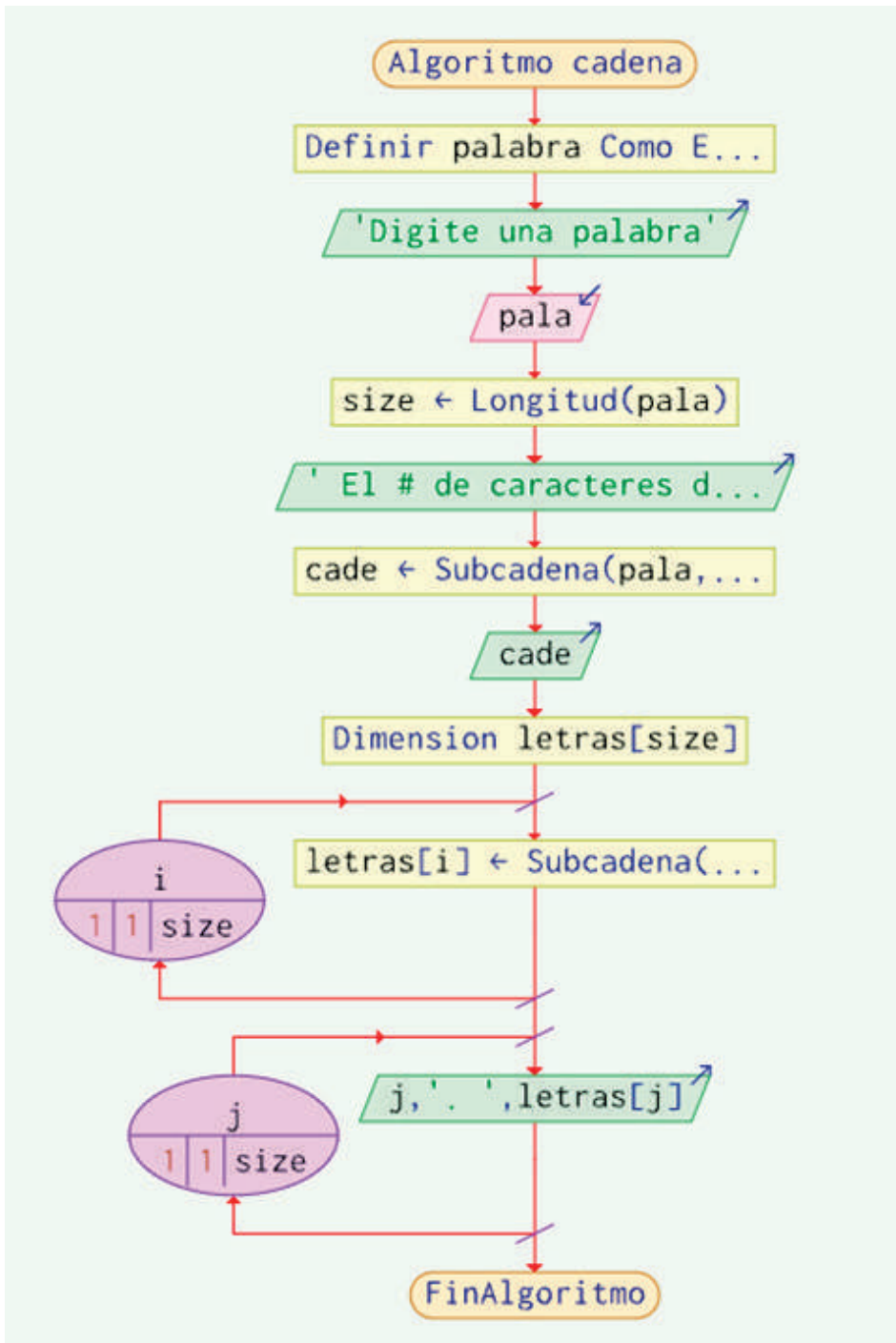
Para j<-1 Hasta size Con Paso 1 Hacer

Escribir j, " ",letras[j]

Fin Para

FinAlgoritmo

7.2.2. Diagrama de flujo



7.2.3. Adaptación a Java

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese una cadena de caracteres:");

        String cadena = sc.nextLine();

        char[] arreglo1 = cadena.toCharArray();

        char[] arreglo2 = new char[arreglo1.length];

        for (int i = 0; i < arreglo1.length; i++) {

            arreglo2[i] = arreglo1[i];

        }

        System.out.println("El contenido del primer arreglo es:" + String.valueOf(arreglo1));

        System.out.println("El contenido del segundo arreglo es:" + String.valueOf(arreglo2));

        sc.close();

    }

}
```

7.3.

Pedir al usuario que digite 2 cadenas de caracteres, e indicar si ambas cadenas son iguales, en caso de no serlo, indicar cuál es mayor alfabéticamente.

7.3.1. Código Pseint

Algoritmo B7_Ejercicio_3

Definir cadena1, cadena2 Como Cadena

Escribir "Ingrese la primera cadena:"

Leer cadena1

Escribir "Ingrese la segunda cadena:"

Leer cadena2

Si cadena1 = cadena2 Entonces

 Escribir "Las cadenas son iguales."

Sino

 Si cadena1 < cadena2 Entonces

 Escribir "La cadena 1 es mayor alfabéticamente."

 Sino

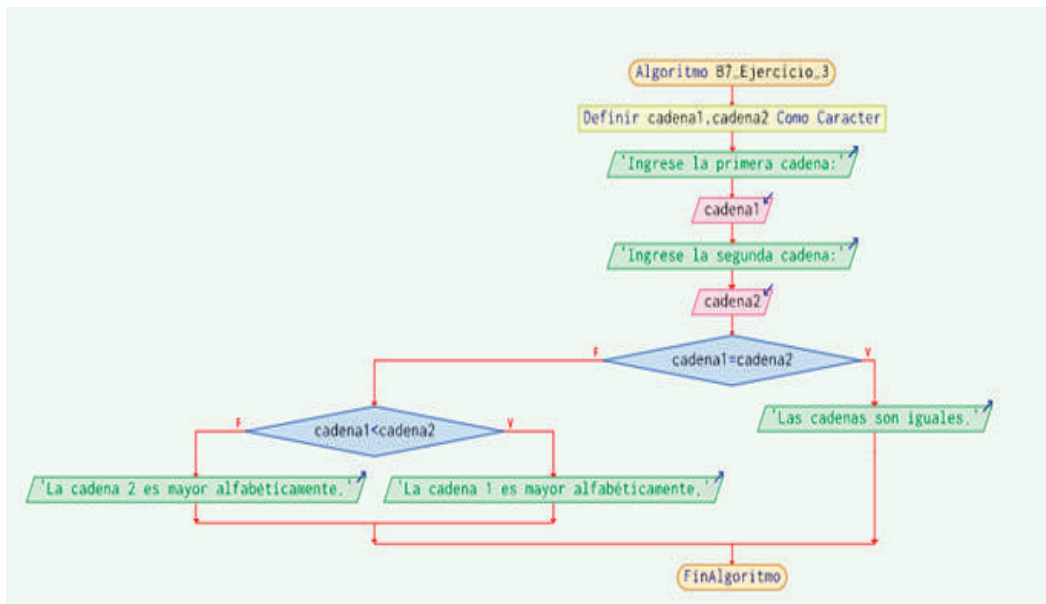
 Escribir "La cadena 2 es mayor alfabéticamente."

 FinSi

FinSi

FinAlgoritmo

7.3.2. Diagrama de flujo



7.3.3. Adaptación a Java

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

```
public class Main{
```

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

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

```
        System.out.println("Ingrese la primera cadena:");
```

```
        String cadena1 = sc.nextLine();
```

```
        System.out.println("Ingrese la segunda cadena:");
```

```
        String cadena2 = sc.nextLine();
```

```
        int comparacion = cadena1.compareToIgnoreCase(cadena2);
```

```
        if (comparacion == 0) {
```

```
            System.out.println("Las dos cadenas son iguales.");
```

```
        } else if (comparacion < 0) {
```

```
            System.out.println("La primera cadena es menor alfabéticamente que la segunda cadena.");
```

```
} else {  
    System.out.println("La primera cadena es mayor alfabéticamente que la  
segunda cadena.");  
}  
sc.close();  
}  
}
```

74.

Crear una cadena que tenga la siguiente frase "Hola que tal", luego crear otra cadena para preguntarle al usuario su nombre, por ultimo añadir el nombre al final de la primera cadena y mostrar el mensaje completo "Hola que tal (NombreDelUsuario)".

74.1. Código Pseint

Algoritmo Saludar

Definir nombre Como Caracter

Escribir "Hola que tal"

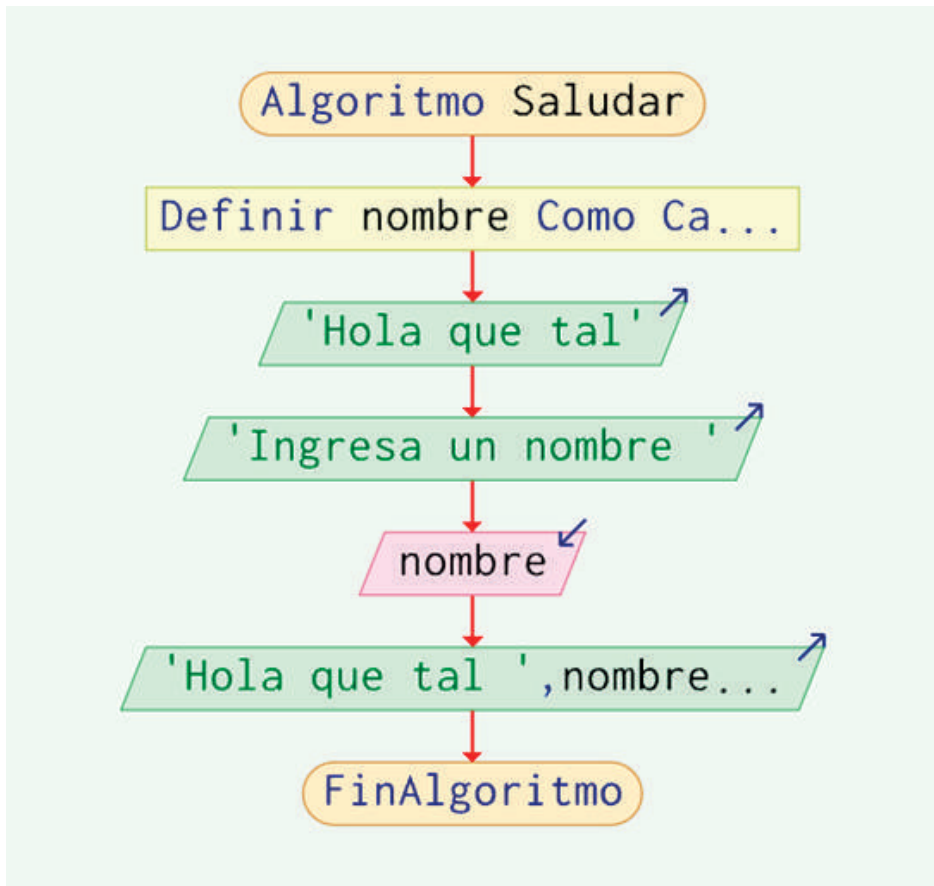
Escribir "Ingresa un nombre "

leer nombre

Escribir "Hola que tal " ,nombre," buenos dias"

FinAlgoritmo

74.2. Diagrama de flujo



74.3. Adaptación a Java

Clase Main:

```
package ejercicioCuatro;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        String nombre = JOptionPane.showInputDialog("Digite su nombre:");
        Cadena frase = new Cadena(nombre);
        frase.MostrarFrase();
    }
}
```

Clase Cadena:

```
package ejercicioCuatro;
public class Cadena {
    String nombre;
    public Cadena(String _nombre){
        nombre = _nombre;
    }
    public void MostrarFrase(){
        System.out.println("Hola que tal " + nombre);
    }
}
```

7.5.

Hacer un programa que determine si una palabra es palíndroma (se lee igual en un sentido que en otro).

7.5.1. Código Pseint

Algoritmo B7_Ejercicio_5

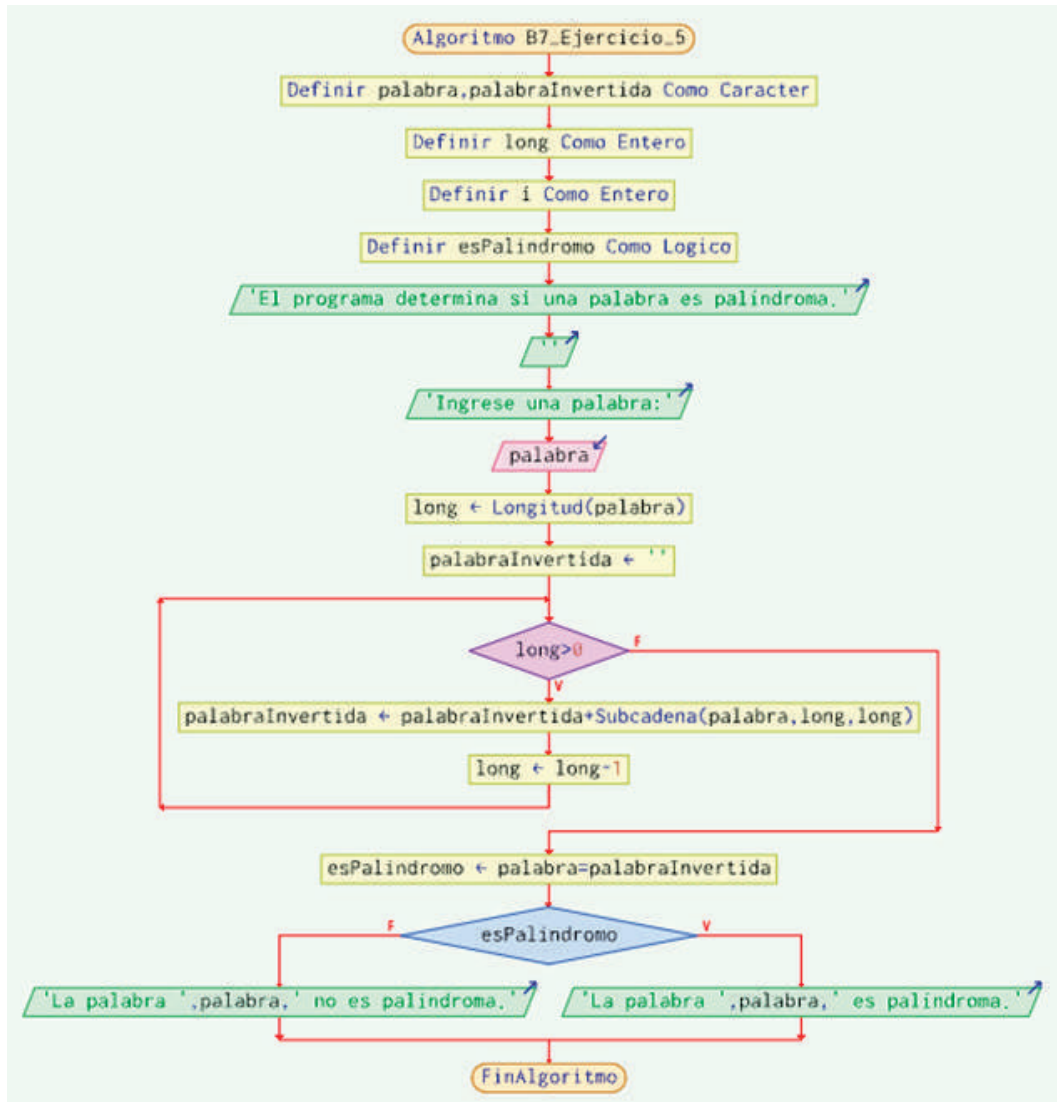
```
Definir palabra, palabraInvertida Como cadena
Definir long Como Entero
Definir i Como Entero

Definir esPalindromo Como Logico

Escribir "El programa determina si una palabra es palíndroma."
Escribir ""
Escribir "Ingrese una palabra:"
Leer palabra
long = Longitud(palabra)
palabraInvertida = ""
Mientras long>0 Hacer
    palabraInvertida= palabraInvertida + Subcadena(palabra,long,long)
    long = long-1
Fin Mientras
esPalindromo <- palabra = palabraInvertida
Si esPalindromo Entonces
    Escribir "La palabra ", palabra, " es palíndroma."
Sino
    Escribir "La palabra ", palabra, " no es palíndroma."
FinSi

FinAlgoritmo
```

7.5.2. Diagrama de flujo



7.5.3. Adaptación a Java

Clase Main:

```
package Palindromos;
import static Palindromos.Palindromo.Verificar;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        String palabra = JOptionPane.showInputDialog("Digite la palabra");
        if (Verificar(palabra) == true){
            System.out.println(palabra + " es palindromo");
        } else{
            System.out.println(palabra + " no es palindromo");
        }
    }
}
```

Clase Palindromo:

```
package Palindromos;
public class Palindromo {
    public static boolean Verificar(String palabra){
        String invertida = new StringBuilder(palabra).reverse().toString();
        return invertida.equals(palabra);
    }
}
```

7.6.

Convertir dos cadenas de minúsculas a MAYUSCULAS. Compararlas, y decir si son iguales o no.

7.6.1. Código Pseint

Algoritmo MayusculasyMinusculas

```
Definir frase, enunciado Como Caracter
Definir x Como Entero
Escribir "Ingresa un texto"

leer frase
Escribir "Ingresa un texto"

leer enunciado
Escribir "Converir a "
Escribir "1 = mayusculas"
Escribir "2 = minusculas"

leer x

si x > 0 y x < 3 Entonces
  si x == 1 Entonces

    Escribir "El texto en mayusculas es:", Mayusculas(frase)
    SiNo
      Escribir "El texto en minusculas es:", Minusculas(frase)

  FinSi
  leer a
  si a == Mayusculas(a) Entonces
    Escribir "El caracter ", a, " son iguales"

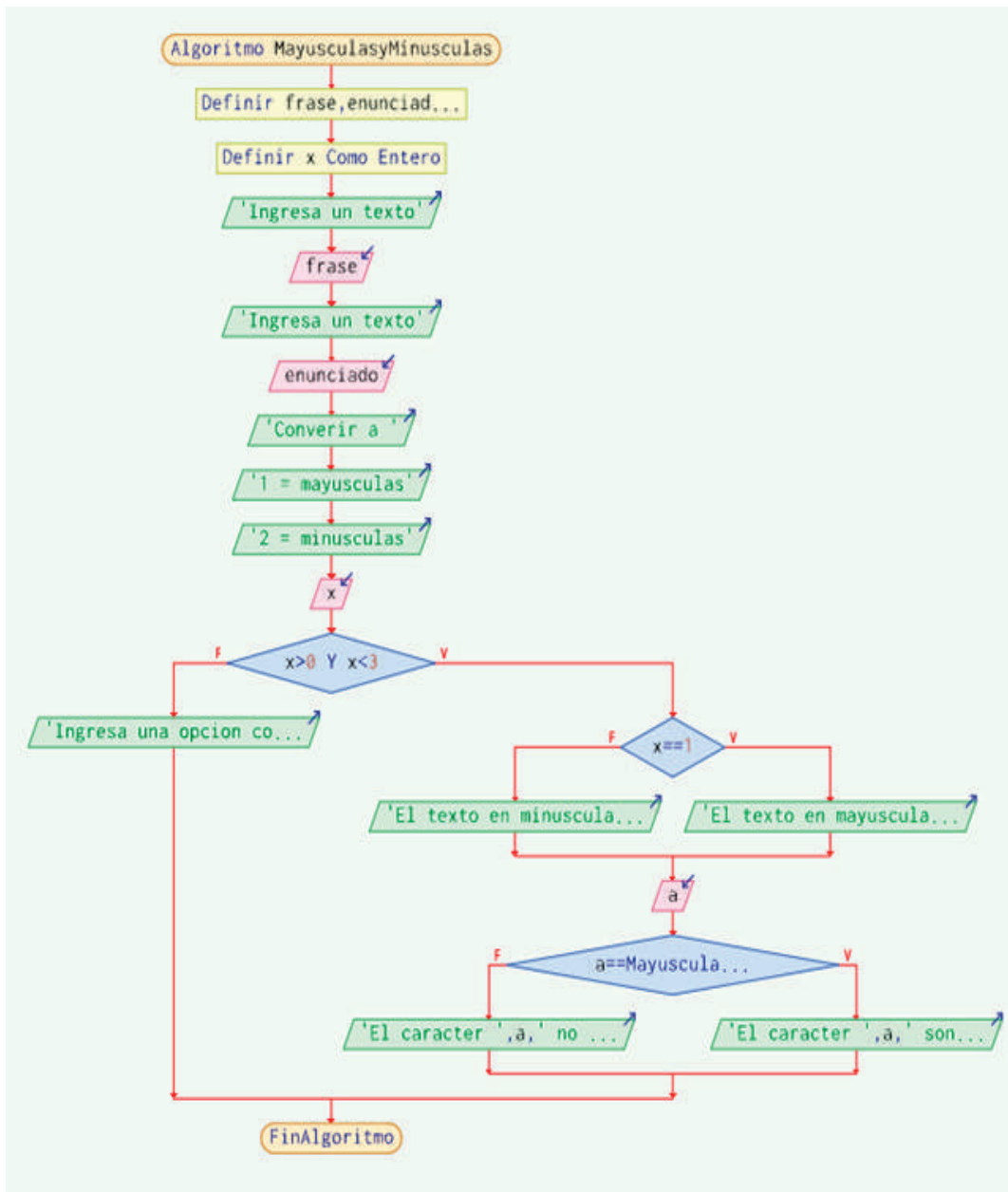
  SiNo
    Escribir "El caracter ", a, " no son iguales"

  FinSi
SiNo
  Escribir "Ingresa una opcion correcta"

FinSi
```

FinAlgoritmo

7.6.2. Diagrama de flujo



7.6.3. Adaptación a Java

```
import javax.swing.JOptionPane;

public class Main{

    public static void main(String[] args) {

        String cadena1, cadena2;

        cadena1 = JOptionPane.showInputDialog("Ingrese la primera cadena").
        toUpperCase();

        cadena2 = JOptionPane.showInputDialog("Ingrese la segunda cadena").
        toUpperCase();

        System.out.println("Las cadenas son:" + cadena1 + " y " + cadena2);

        boolean sonIguales = cadena1.equals(cadena2);

        if (sonIguales) {

            JOptionPane.showMessageDialog(null, "Las cadenas son iguales");

        } else {

            JOptionPane.showMessageDialog(null, "Las cadenas no son iguales");

        }

    }

}
```

7.7.

Pedir su nombre al usuario en MAYÚSCULA, si su nombre comienza por la letra A, convertir su nombre a minúsculas, caso contrario no convertirlo.

7.7.1. Código Pseint

Algoritmo B7_Ejercicio_7

Definir nombre, nombreConvertido Como Cadena

Escribir “El programa pide su nombre al usuario en MAYÚSCULA, si su nombre comienza por la letra A, convierte su nombre a minúsculas, caso contrario no lo convierte.”

Escribir “”

Repetir

Escribir “Ingrese su nombre en MAYÚSCULA:”

Leer nombre

Si Minusculas(nombre) = nombre Entonces

Escribir “El nombre debe estar en MAYÚSCULA. Por favor, ingréselo nuevamente.”

FinSi

Hasta Que Minusculas(nombre) <> nombre

Si SubCadena(nombre, 1, 1) = “A” Entonces

nombreConvertido <- Minusculas(nombre)

Sino

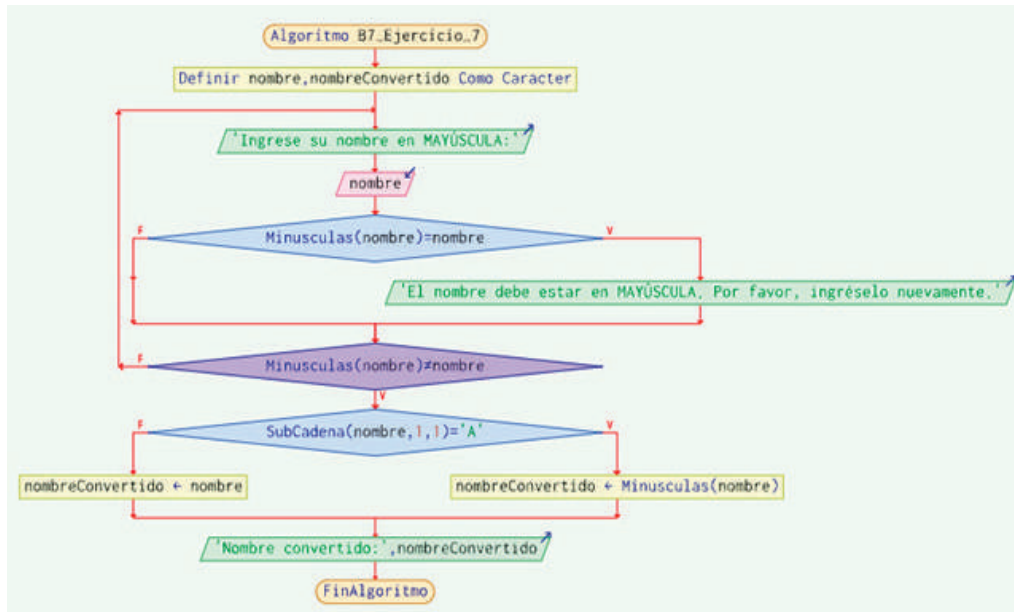
nombreConvertido <- nombre

FinSi

Escribir “Nombre convertido:”, nombreConvertido

FinAlgoritmo

7.7.2. Diagrama de flujo



7.7.3. Adaptación a Java

```
import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {

        String nombre;

        nombre = JOptionPane.showInputDialog("Ingrese su nombre en MAYÚSCULAS");

        if (nombre.startsWith("A")) {
            nombre = nombre.toLowerCase();

            JOptionPane.showMessageDialog(null, "Nombre en minúsculas:" + nombre);
        } else {
            JOptionPane.showMessageDialog(null, "El nombre no empieza por la letra A");
        }
    }
}
```

7.8.

Pedir al usuario 2 cadenas de caracteres de números, uno entero y el otro real, convertirlos a sus respectivos valores y por último sumarlos.

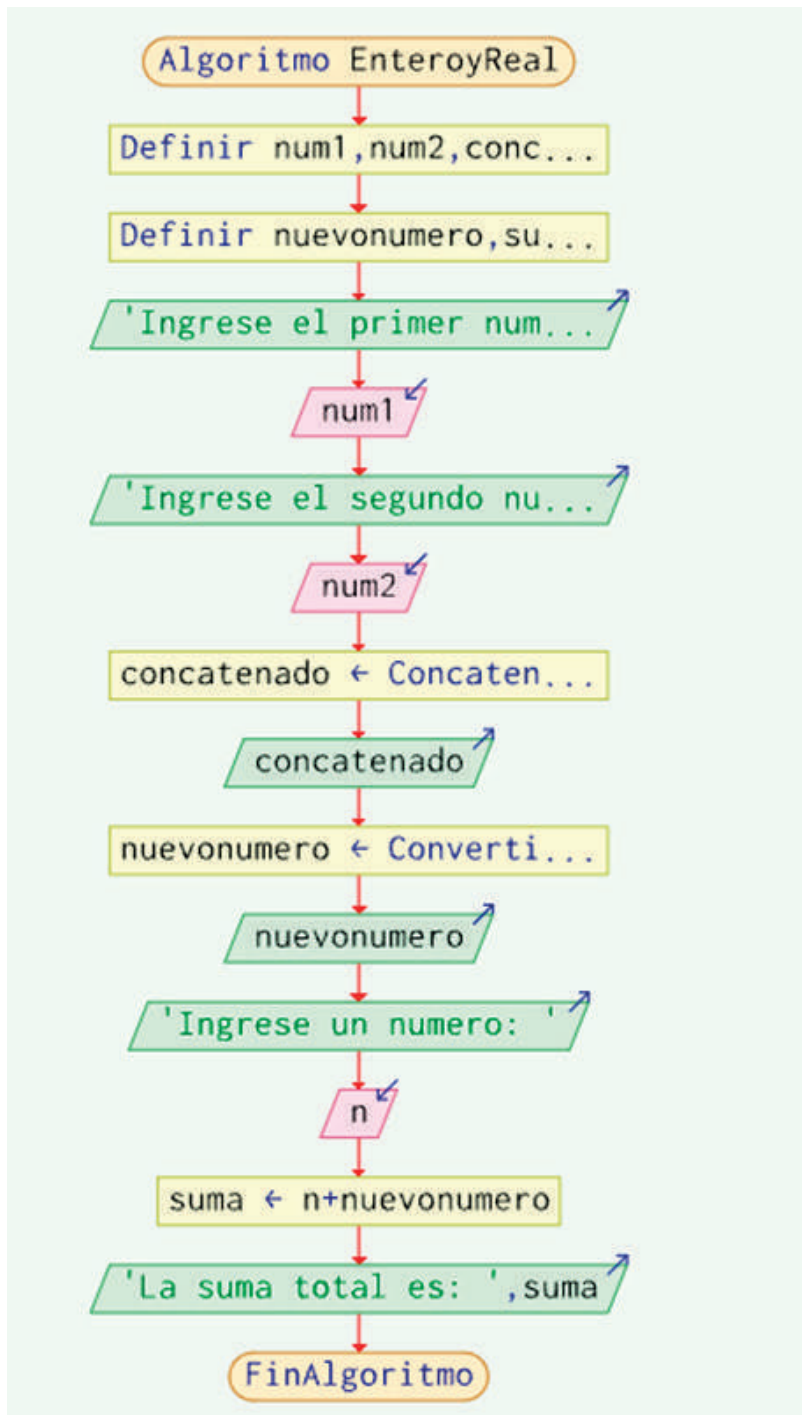
7.8.1. Código Pseint

Algoritmo EnteroyReal

```
Definir num1, num2, concatenado Como Caracter;  
Definir nuevonumero,suma,n Como Entero;  
Escribir "Ingrese el primer numero:";  
leer num1;  
Escribir "Ingrese el segundo numero:";  
leer num2;  
concatenado<-Concatenar(num1,num2)  
Escribir concatenado;  
nuevonumero<-ConvertirANumero(concatenado);  
Escribir nuevonumero;  
Escribir "Ingrese un numero:";  
leer n;  
suma<-n+nuevonumero;  
Escribir "La suma total es:", suma;
```

FinAlgoritmo

7.8.2. Diagrama de flujo



7.8.3. Adaptación a Java

```
import javax.swing.JOptionPane;
public class Main{
    public static void main(String[] args) {
        String enteros = JOptionPane.showInputDialog("Ingrese su numero entero");
        String reales = JOptionPane.showInputDialog("Ingrese su numero real");
        int Enteros = Integer.parseInt(enteros);
        double Reales = Double.parseDouble(reales);
        double suma = Enteros+Reales;
        JOptionPane.showMessageDialog(null,"La suma de sus numeros es "+suma);
    }
}
```

7.9.

Realice un programa que lea una cadena de caracteres de la entrada estándar y muestre en la salida estándar cuántas ocurrencias de cada vocal existen en la cadena.

7.9.1. Código Pseint

Algoritmo B7_Ejercicio_9

Definir cadena Como cadena
Definir CHARACTER Como Caracter
Definir contadorA,contadorE,contadorI,contadorO,contadorU Como Entero
Escribir 'Ingrese una cadena de caracteres:'

Leer cadena
contadorA <- 0
contadorE <- 0
contadorI <- 0
contadorO <- 0
contadorU <- 0

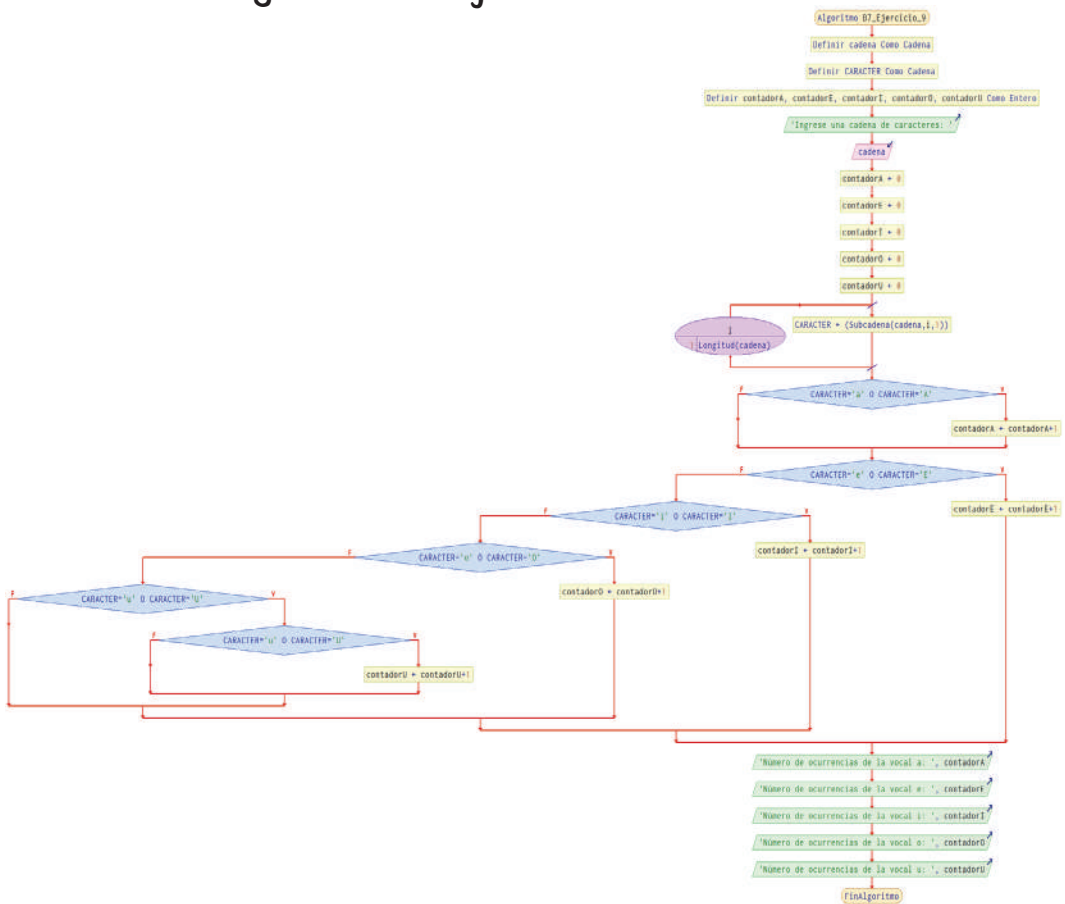
```

Para i<-1 Hasta Longitud(cadena) Hacer
  CARACTER <- (Subcadena(cadena,i,l))
  Si CARACTER='a' O CARACTER='A' Entonces
    contadorA <- contadorA+1
  SiNo
    Si CARACTER='e' O CARACTER='E' Entonces
      contadorE <- contadorE+1
    SiNo
      Si CARACTER='i' O CARACTER='I' Entonces
        contadorI <- contadorI+1
      SiNo
        Si CARACTER='o' O CARACTER='O' Entonces
          contadorO <- contadorO+1
        SiNo
          Si CARACTER='u' O CARACTER='U' Entonces
            SiNo
              Si CARACTER='u' O CARACTER='U' Entonces
                contadorU <- contadorU+1
          FinSi
        FinSi
      FinSi
    FinSi
  FinSi
FinPara
Escribir 'Número de ocurrencias de la vocal a:',contadorA
Escribir 'Número de ocurrencias de la vocal e:',contadorE
Escribir 'Número de ocurrencias de la vocal i:',contadorI
Escribir 'Número de ocurrencias de la vocal o:',contadorO
Escribir 'Número de ocurrencias de la vocal u:',contadorU

```

FinAlgoritmo

7.9.2. Diagrama de flujo



7.9.3. Adaptación a Java

```
import javax.swing.JOptionPane;
```

```
public class Main{
```

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

```
        String cadena = JOptionPane.showInputDialog("Ingrese su cadena de caracteres").toLowerCase(); // Convertir la cadena a minúsculas
```

```
        // creamos un contador para cada vocal
```

```
        int contadorA = 0;
```

```
        int contadorE = 0;
```

```
        int contadorI = 0;
```

```
        int contadorO = 0;
```

```
int contadorU = 0;

// Recorrer la cadena y contar las ocurrencias de cada vocal
for (int i = 0; i < cadena.length(); i++) {

    char caracter = cadena.charAt(i);

    switch (caracter) {
        case 'a':

            contadorA++;
            break;

        case 'e':
            contadorE++;
            break;

        case 'i':
            contadorI++;
            break;

        case 'o':
            contadorO++;
            break;

        case 'u':
            contadorU++;
            break;
    }
}

// Mostramos el resultado

System.out.println("Cantidad de ocurrencias de cada vocal:");
System.out.println("A:" + contadorA);
System.out.println("E:" + contadorE);
System.out.println("I:" + contadorI);
System.out.println("O:" + contadorO);
System.out.println("U:" + contadorU);
}
```

Bloque 08: Estructuras

8.1.

Hacer una estructura llamada corredor, en la cual se tendrán los siguientes campos: Nombre, edad, sexo, club, pedir datos al usuario para un corredor, y asignarle una categoría de competición:

Juvenil \leq 18 años

Señor \leq 40 años

Veterano $>$ 40 años

Posteriormente imprimir todos los datos del corredor, incluida su categoría de competición.

8.1.1. Código Pseint

Algoritmo B8_Ejercicio_I

Definir nombre, club, categoria como cadena

Definir edad como entero

Definir sexo como caracter

Escribir "Ingrese el nombre del corredor:"

Leer nombre

Escribir "Ingrese la edad del corredor:"

Leer edad

Escribir "Ingrese el sexo del corredor (M/F):"

Leer sexo

Escribir "Ingrese el club del corredor:"

Leer club

Si edad \leq 18 Entonces

 categoria = "Juvenil"

SiNo

 Si edad \leq 40 Entonces

 categoria = "Señor"

 SiNo

 categoria = "Veterano"

 Fin Si

Fin Si

Escribir "Datos del corredor:"

Escribir "Nombre:" + nombre

Escribir "Edad:", edad

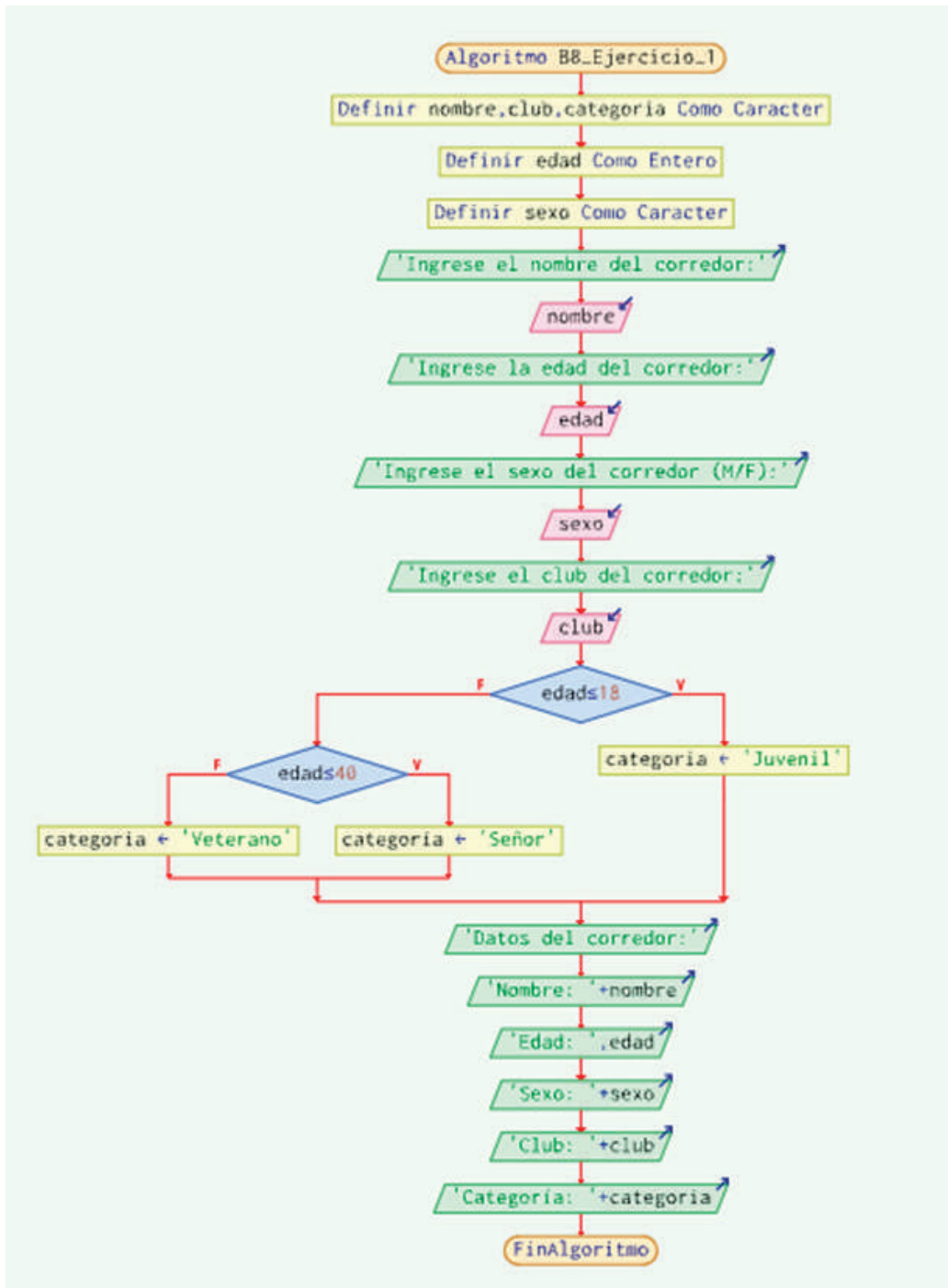
Escribir "Sexo:" + sexo

Escribir "Club:" + club

Escribir "Categoría:" + categoria

FinAlgoritmo

8.1.2. Diagrama de flujo



8.1.3. Adaptación a Java

```
import java.util.Scanner;
public final class Corredor {
    private String nombre;
    private int edad;
    private String sexo;
    private String club;
    private String categoria;
    public Corredor(String nombre, int edad, String sexo, String club) {
        this.nombre = nombre;
        this.edad = edad;
        this.sexo = sexo;
        this.club = club;
        this.categoria = asignarCategoria(edad);
    }
    public String asignarCategoria(int edad) {
        if (edad <= 18) {
            return "Juvenil";
        } else if (edad <= 40) {
            return "Señor";
        } else {
            return "Veterano";
        }
    }
}
public void imprimirDatos() {
```

```
System.out.println("Nombre:" + nombre);
    System.out.println("Edad:" + edad);
    System.out.println("Sexo:" + sexo);
    System.out.println("Club:" + club);
    System.out.println("Categoría:" + categoria);
}
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Ingrese el nombre del corredor:");
    String nombre = scanner.nextLine();
    System.out.print("Ingrese la edad del corredor:");
    int edad = scanner.nextInt();
    scanner.nextLine(); // Consumir el salto de línea después de leer un entero
    System.out.print("Ingrese el sexo del corredor:");
    String sexo = scanner.nextLine();
    System.out.print("Ingrese el club del corredor:");
    String club = scanner.nextLine();
    Corredor corredor = new Corredor(nombre, edad, sexo, club);
    System.out.println("\nDatos del corredor:");
    corredor.imprimirDatos();
}
}
```

8.2.

Hacer una estructura llamada alumno, en la cual se tendrán los siguientes Campos: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.

8.2.1. Código Pseint

Algoritmo alumnos

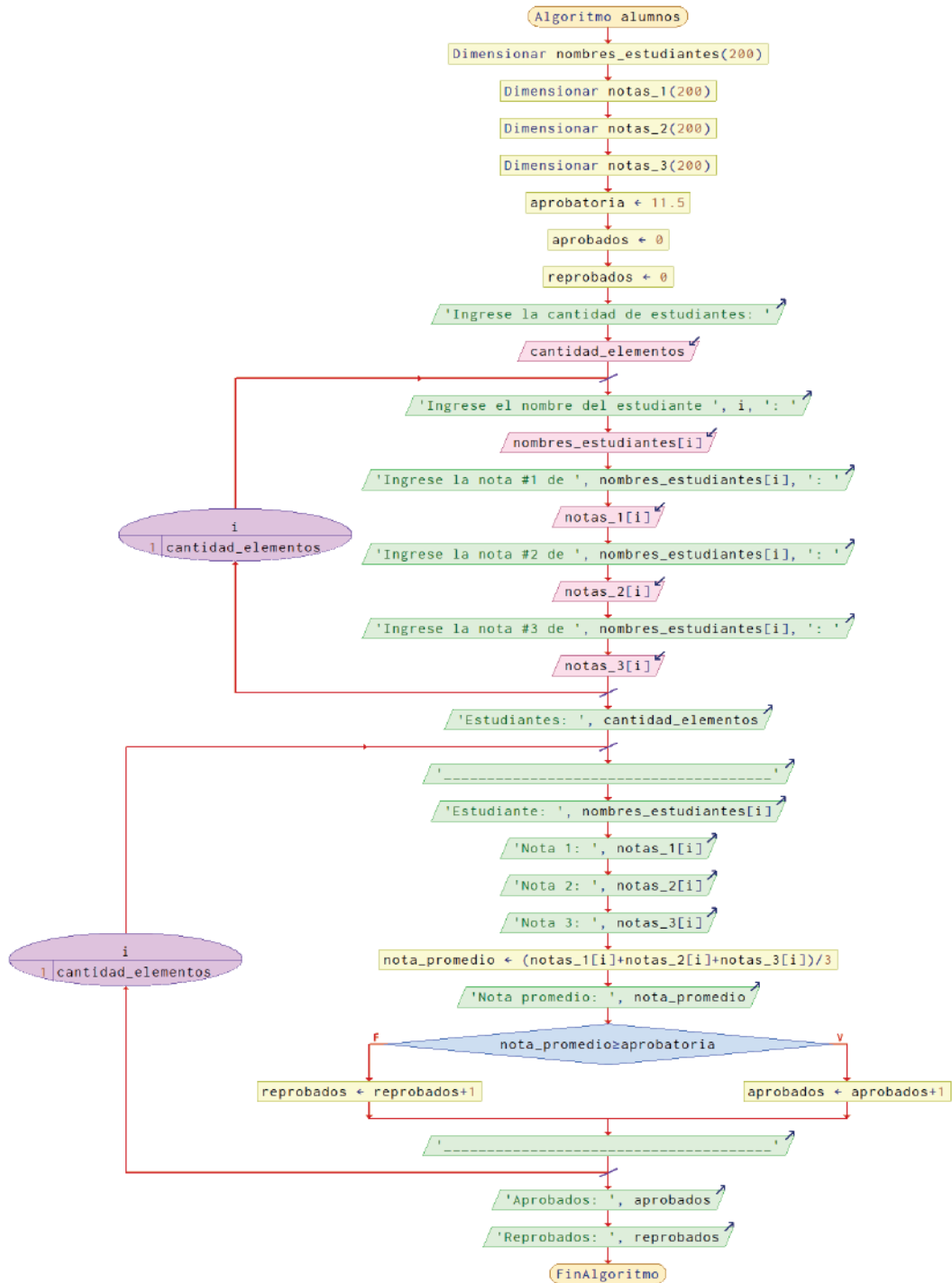
```
Dimension nombres_estudiantes[200];
Dimension notas_1[200];
Dimension notas_2[200];
Dimension notas_3[200];
aprobatoria <- 11.5
aprobados <- 0
reprobados <- 0
Escribir "Ingrese la cantidad de estudiantes:";
Leer cantidad_elementos;
Para i<-1 Hasta cantidad_elementos Hacer
    Escribir "Ingrese el nombre del estudiante ";
    Leer nombres_estudiantes[i];
    Escribir "Ingrese la nota #1 de ";
    Leer notas_1[i];
    Escribir "Ingrese la nota #2 de ";
    Leer notas_2[i];
    Escribir "Ingrese la nota #3 de ";
    Leer notas_3[i];
FinPara
```



```
Escribir "Estudiantes:", cantidad_elementos;
Para i<-1 Hasta cantidad_elementos Hacer
    Escribir "_____";
    Escribir "Estudiante:", nombres_estudiantes[i];
    Escribir "Nota 1:", notas_1[i];
    Escribir "Nota 2:", notas_2[i];
    Escribir "Nota 3:", notas_3[i];
    nota_promedio <- (notas_1[i]+notas_2[i]+notas_3[i])/3;
    Escribir "Nota promedio:", nota_promedio;
    Si nota_promedio >= aprobatoria Entonces
        aprobados = aprobados + 1
    SiNo
        reprobados = reprobados + 1
    FinSi;
    Escribir "_____";
FinPara
Escribir "Aprobados:", aprobados;
Escribir "Reprobados:", reprobados;
```

FinAlgoritmo

8.2.2. Diagrama de flujo



8.2.3. Adaptación a Java

```
import java.util.Scanner;

public class Alumno {
    private String nombre;
    private int edad;
    private double promedio;

    public Alumno(String nombre, int edad, double promedio) {
        this.nombre = nombre;
        this.edad = edad;
        this.promedio = promedio;
    }

    public String getNombre() {
        return nombre;
    }

    public int getEdad() {
        return edad;
    }

    public double getPromedio() {
        return promedio;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Alumno mejorAlumno = null;
        double mejorPromedio = Double.MIN_VALUE;
        for (int i = 1; i <= 3; i++) {
```

```
System.out.println("Datos del alumno " + i + ":");
System.out.print("Nombre:");
String nombre = scanner.nextLine();
System.out.print("Edad:");
int edad = Integer.parseInt(scanner.nextLine());
System.out.print("Promedio:");
double promedio = Double.parseDouble(scanner.nextLine());
Alumno alumno = new Alumno(nombre, edad, promedio);
if (promedio > mejorPromedio) {
    mejorPromedio = promedio;
    mejorAlumno = alumno;
}
System.out.println();
}
System.out.println("Alumno con mejor promedio:");
System.out.println("Nombre:" + mejorAlumno.getNombre());
System.out.println("Edad:" + mejorAlumno.getEdad());
System.out.println("Promedio:" + mejorAlumno.getPromedio());
}
}
```

8.3.

Realizar un programa que lea un arreglo de estructuras los datos de N empleados de la empresa y que imprima los datos del empleado con mayor y menor salario.

8.3.1. Código Pseint

Algoritmo B8_Ejercicio_3

Definir n, i, salario, salarioMaximo, salarioMinimo como Real

Definir nombreMaximo, nombreMinimo como Cadena

Escribir "Ingrese el número de empleados:"

Leer n

salarioMaximo <- 0

salarioMinimo <- 0

Para i <- 1 Hasta n Hacer

Escribir "Ingrese el nombre del empleado ", i, ":"

Leer nombreMaximo

Escribir "Ingrese el salario del empleado ", i, ":"

Leer salario

Si i = 1 Entonces

nombreMinimo <- nombreMaximo

salarioMinimo <- salario

Sino

Si salario > salarioMaximo Entonces

salarioMaximo <- salario

nombreMaximo <- nombreMaximo

FinSi

Si salario < salarioMinimo Entonces

salarioMinimo <- salario

nombreMinimo <- nombreMaximo

FinSi

FinSi

FinPara

Escribir "Datos del empleado con mayor salario:"

Escribir "Nombre:", nombreMaximo

Escribir "Salario:", salarioMaximo

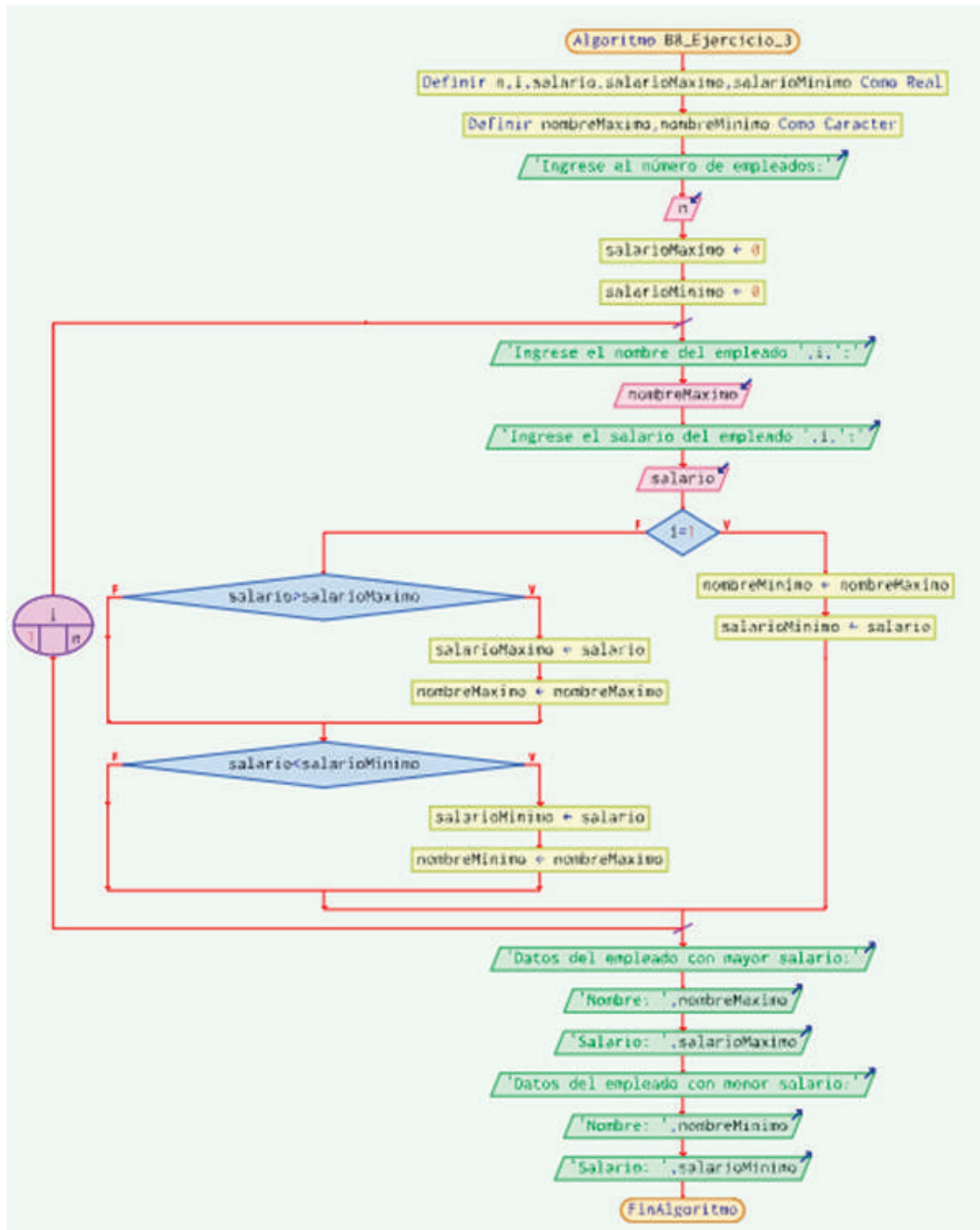
Escribir "Datos del empleado con menor salario:"

Escribir "Nombre:", nombreMinimo

Escribir "Salario:", salarioMinimo

FinAlgoritmo

8.3.2. Diagrama de flujo



8.3.3. Adaptación a Java

```
import java.util.Scanner;

public class Estructura3 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el número de empleados:");

        int n = sc.nextInt();

        Empleado[] empleados = new Empleado[n];

        for (int i = 0; i < n; i++) {

            System.out.println("Ingrese los datos del empleado " + (i + 1) + ":");

            System.out.print("Nombre:");

            String nombre = sc.next();

            System.out.print("Salario:");

            double salario = sc.nextDouble();

            empleados[i] = new Empleado(nombre, salario);

        }

        int indiceMayorSalario = 0;

        int indiceMenorSalario = 0;

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

            if (empleados[i].getSalario() > empleados[indiceMayorSalario].getSalario()) {

                indiceMayorSalario = i;

            }

            if (empleados[i].getSalario() < empleados[indiceMenorSalario].getSalario()) {

                indiceMenorSalario = i;

            }

        }

    }

}
```

```
System.out.println("\nEmpleado con mayor salario:");
System.out.println("Nombre:" + empleados[indiceMayorSalario].getNombre());
System.out.println("Salario:" + empleados[indiceMayorSalario].getSalario());
System.out.println("\nEmpleado con menor salario:");
System.out.println("Nombre:" + empleados[indiceMenorSalario].getNombre());
System.out.println("Salario:" + empleados[indiceMenorSalario].getSalario());
sc.close();
}
}
class Empleado {
    private final String nombre;
    private final double salario;
    public Empleado(String nombre, double salario) {
        this.nombre = nombre;
        this.salario = salario;
    }
    public String getNombre() {
        return nombre;
    }
    public double getSalario() {
        return salario;
    }
}
```


84.

Hacer un arreglo de estructura llamada atleta para N atletas que contenga los siguientes campos: nombre, país, número de medallas. Y devuelva los datos (Nombre, país) del atleta que ha ganado el mayor número de medallas.

84.1. Código Pseint

Algoritmo medallero

Escribir “Cuál es su nombre ?”

leer nombre

Escribir “En qué posición terminó?”

leer posicion

Segun posicion Hacer

1:

Escribir “Gano medalla de oro”

2:

Escribir “Gano medalla de plata”

3:

Escribir “Gano medalla de bronce”

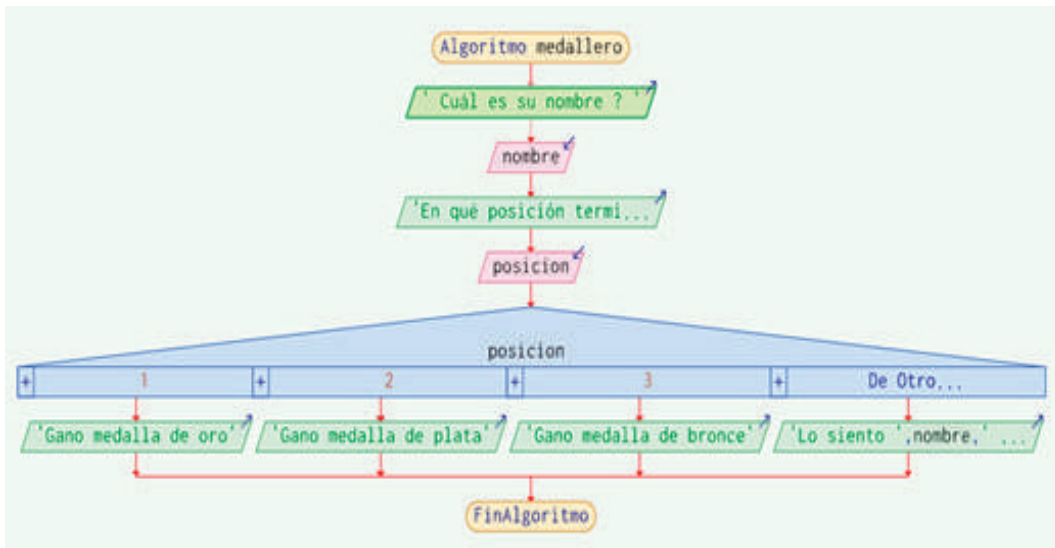
De Otro Modo:

Escribir “Lo siento “, nombre,“ no gano nada “

FinSegun

FinAlgoritmo

84.2. Diagrama de flujo



84.3. Adaptación a Java

```

public class Estructura4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el número de atletas:");
        int n = sc.nextInt();
        Atleta[] atletas = new Atleta[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Ingrese los datos del atleta " + (i + 1) + ":",);
            System.out.print("Nombre:");
            String nombre = sc.next();
            System.out.print("País:");
            String pais = sc.next();
            System.out.print("Número de medallas:");
            int numMedallas = sc.nextInt();
        }
    }
}
  
```

```
        atletas[i] = new Atleta(nombre, pais, numMedallas);
    }

    int indiceMayorMedallas = 0;
    for (int i = 1; i < n; i++) {

        if (atletas[i].getNumMedallas() > atletas[indiceMayorMedallas].getNumMedallas()) {
            indiceMayorMedallas = i;
        }
    }

    System.out.println("\nAtleta con mayor n m
    System.out.println("\nAtleta con mayor n mero de medallas:");
    System.out.println("Nombre: " + atletas[indiceMayorMedallas].getNombre());
    System.out.println("Pa s: " + atletas[indiceMayorMedallas].getPais());

    sc.close();
}
}

class Atleta {

    private final String nombre;
    private final String pais;
    private final int numMedallas;
    public Atleta(String nombre, String pais, int numMedallas) {

        this.nombre = nombre;
        this.pais = pais;
        this.numMedallas = numMedallas;
    }
    public String getNombre() {
        return nombre;
    }

    public String getPais() {
        return pais;
    }

    public int getNumMedallas() {
        return numMedallas;
    }
}
```

8.5.

Hacer 2 estructuras una llamada promedio que tendrá los siguientes campos: nota1, nota2, nota3; y otro llamada alumno que tendrá los siguientes miembros: nombre, sexo, edad; hacer que la estructura promedio este anidada en la estructura alumno, luego pedir todos los datos para un alumno, luego calcular su promedio, y por último imprimir todos sus datos incluidos el promedio.

8.5.1. Código Pseint

Algoritmo B8_Ejercicio_5

Definir nombre como Cadena

Definir sexo como Caracter

Definir edad como Entero

Definir nota1, nota2, nota3 como Real

Escribir "Ingrese el nombre del alumno:"

Leer nombre

Escribir "Ingrese el sexo del alumno:"

Leer sexo

Escribir "Ingrese la edad del alumno:"

Leer edad

Escribir "Ingrese las tres notas del alumno:"

Leer nota1

Leer nota2

Leer nota3

Definir promedioFinal como Real

promedioFinal <- (nota1 + nota2 + nota3) / 3

Escribir "Datos del alumno:"

Escribir "Nombre:", nombre

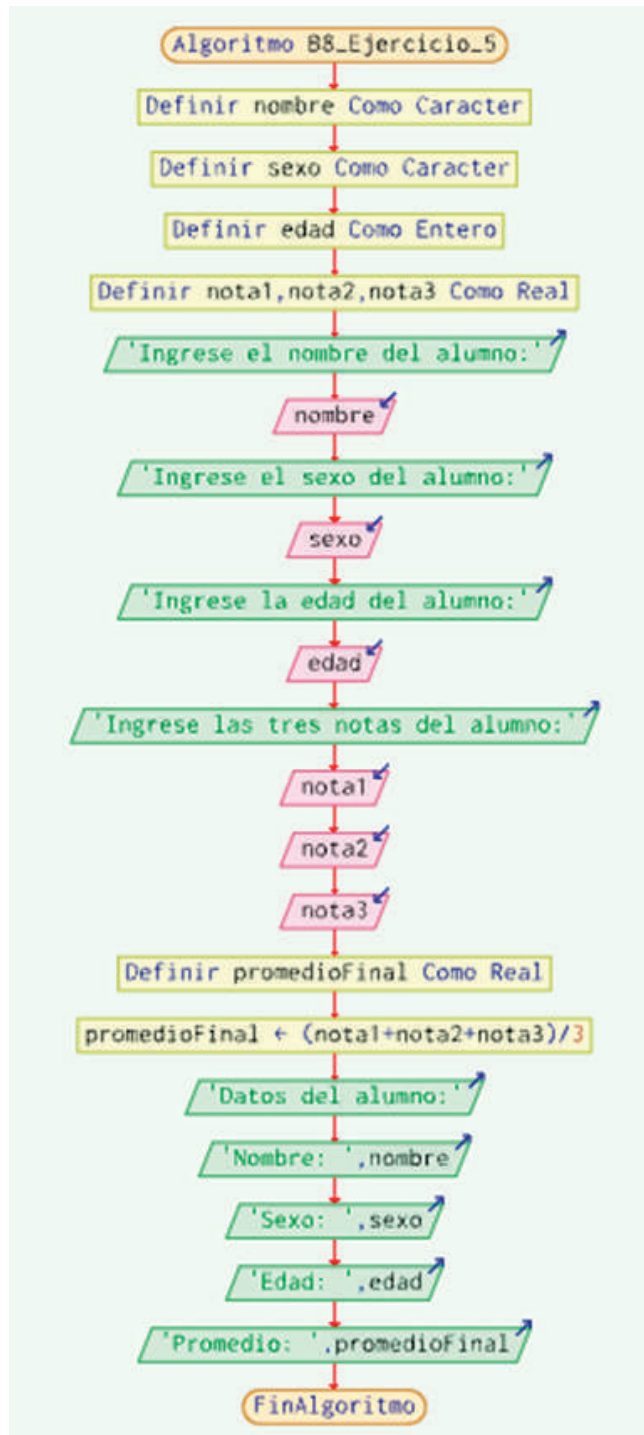
Escribir "Sexo:", sexo

Escribir "Edad:", edad

Escribir "Promedio:", promedioFinal

FinAlgoritmo

8.5.2. Diagrama de flujo



8.5.3. Adaptación a Java

```
package Cadenas;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        class Promedio {
            double nota1;
            double nota2;
            double nota3;
        }

        class Alumno {

            String nombre;
            String sexo;
            int edad;
            Promedio promedio;
        }

        Alumno alumno = new Alumno();

        System.out.println("Ingrese los datos del alumno:");

        System.out.print("Nombre:");

        alumno.nombre = scanner.nextLine();

        System.out.print("Sexo:");

        alumno.sexo = scanner.nextLine();

        System.out.print("Edad:");

        alumno.edad = scanner.nextInt();

        System.out.println("Ingrese las notas del alumno:");

        alumno.promedio = new Promedio();

        System.out.print("Nota 1:");

        alumno.promedio.nota1 = scanner.nextDouble();
```

```
System.out.print("Nota 2:");
alumno.promedio.nota2 = scanner.nextDouble();

System.out.print("Nota 3:");

alumno.promedio.nota3 = scanner.nextDouble();

double promedioFinal = (alumno.promedio.nota1 + alumno.promedio.nota2 +
alumno.promedio.nota3) / 3;

System.out.println("\nInformacion del alumno:");

System.out.println("Nombre:" + alumno.nombre);

System.out.println("Sexo:" + alumno.sexo);

System.out.println("Edad:" + alumno.edad);

System.out.println("Promedio:" + promedioFinal);

scanner.close();
}
}
```

8.6.

Utilizar las 2 estructuras del problema 5, pero ahora pedir los datos para N alumnos, y calcular cuál de todos tiene el mejor promedio, e imprimir sus datos.

8.6.1. Código Pseint

Algoritmo N datos

```
Dimension nombres_estudiantes[200];
Dimension notas_1[200];
Dimension notas_2[200];
Dimension notas_3[200];

aprobatoria <- 10.5
aprobados <- 0
reprobados <- 0

Escribir "Ingrese la cantidad de estudiantes:";

Leer cantidad_elementos;
```

```

Para i<-1 Hasta cantidad_elementos Hacer
  Escribir "Ingrese el nombre del estudiante ",i,";"
  Leer nombres_estudiantes[i];
  Escribir "Ingrese la nota #1 de ",nombres_estudiantes[i],";"
  Leer notas_1[i];
  Escribir "Ingrese la nota #2 de ",nombres_estudiantes[i],";"
  Leer notas_2[i];
  Escribir "Ingrese la nota #3 de ",nombres_estudiantes[i],";"
  Leer notas_3[i];
FinPara
Escribir "Estudiantes:", cantidad_elementos;

Para i<-1 Hasta cantidad_elementos Hacer
  Escribir "_____";

  Escribir "Estudiante:", nombres_estudiantes[i];
  Escribir "Nota 1:", notas_1[i];
  Escribir "Nota 2:", notas_2[i];
  Escribir "Nota 3:", notas_3[i];
  nota_promedio <- (notas_1[i]+notas_2[i]+notas_3[i])/3;

  Escribir "Nota promedio:", nota_promedio;
  Si nota_promedio >= aprobatoria Entonces
    aprobados = aprobados + 1

  SiNo
    reprobados = reprobados + 1

  FinSi;
  Escribir "_____";

FinPara
Escribir "Aprobados:", aprobados;
Escribir "Reprobados:", reprobados;

```

FinAlgoritmo

8.6.3. Adaptación a Java

```
import java.util.Scanner;

public class Main {
    static class Promedio {
        double nota1;
        double nota2;
        double nota3;
    }
    static class Alumno {

        String nombre;
        String sexo;
        int edad;
        Promedio promedio;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese la cantidad de alumnos:");

        int cantidadAlumnos = scanner.nextInt();

        Alumno[] alumnos = new Alumno[cantidadAlumnos]

        for (int i = 0; i < cantidadAlumnos; i++) {
            Alumno alumno = new Alumno();

            System.out.println("\nIngrese los datos del alumno " + (i + 1) + " :");

            System.out.print("Nombre:");

            scanner.nextLine(); // Limpiar el buffer
            alumno.nombre = scanner.nextLine();

            System.out.print("Sexo:");

            alumno.sexo = scanner.nextLine();

            System.out.print("Edad:");

            alumno.edad = scanner.nextInt();

            System.out.println("Ingrese las notas del alumno " + (i + 1) + " :");

            alumno.promedio = new Promedio();
```

```
System.out.print("Nota 1:");
    alumno.promedio.nota1 = scanner.nextDouble();
    System.out.print("Nota 2:");
    alumno.promedio.nota2 = scanner.nextDouble();
    System.out.print("Nota 3:");
    alumno.promedio.nota3 = scanner.nextDouble();
    alumnos[i] = alumno;
}
int indiceMejorPromedio = 0;
double mejorPromedio = calcularPromedio(alumnos[0].promedio);
for (int i = 1; i < cantidadAlumnos; i++) {
    double promedioActual = calcularPromedio(alumnos[i].promedio);
    if (promedioActual > mejorPromedio) {
        mejorPromedio = promedioActual;
        indiceMejorPromedio = i;
    }
}
System.out.println("\nDatos del alumno con el mejor promedio:");
System.out.println("Nombre:" + alumnos[indiceMejorPromedio].nombre);
System.out.println("Sexo:" + alumnos[indiceMejorPromedio].sexo);
System.out.println("Edad:" + alumnos[indiceMejorPromedio].edad);
System.out.println("Promedio:" + mejorPromedio);
scanner.close();
}
private static double calcularPromedio(Promedio promedio) {
    return (promedio.nota1 + promedio.nota2 + promedio.nota3) / 3;
}
}
```

8.7.

Defina una estructura que indique el tiempo empleado por un ciclista en una etapa. La estructura debe tener tres campos: horas, minutos y segundos. Escriba un programa que dado n etapas calcule el tiempo total empleado en correr todas las etapas.

8.7.1. Código Pseint

Algoritmo B8_Ejercicio_7

```
Definir n, horasTotal, minutosTotal, segundosTotal como Entero
Definir horas, minutos, segundos como Entero
Escribir "Ingrese el número de etapas:"
```

```
Leer n
horasTotal <- 0
minutosTotal <- 0
segundosTotal <- 0
```

```
Para i <- 1 Hasta n Hacer
    Escribir "Ingrese el tiempo empleado en la etapa ", i, ":"
    Escribir "Horas:"
    Leer horas
    Escribir "Minutos:"
```

```
    Leer minutos
    Escribir "Segundos:"
```

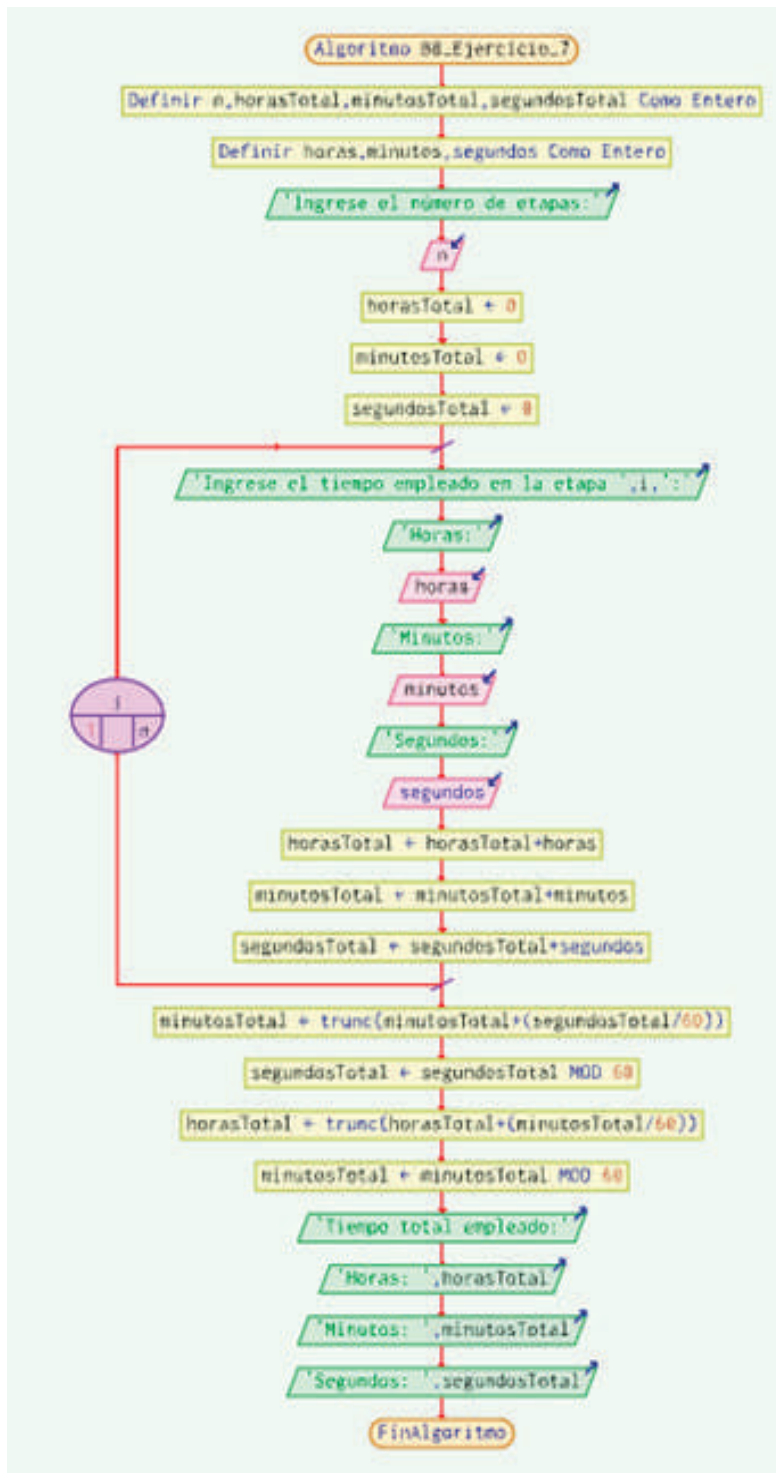
```
    Leer segundos
    horasTotal <- horasTotal + horas
    minutosTotal <- minutosTotal + minutos
    segundosTotal <- segundosTotal + segundos
```

```
FinPara
```

```
minutosTotal <- trunc(minutosTotal + (segundosTotal / 60))
segundosTotal <- segundosTotal mod 60
horasTotal <- trunc(horasTotal + (minutosTotal / 60))
minutosTotal <- minutosTotal mod 60
Escribir "Tiempo total empleado:"
Escribir "Horas:", horasTotal
Escribir "Minutos:", minutosTotal
Escribir "Segundos:", segundosTotal
```

FinAlgoritmo

8.7.2. Diagrama de flujo



8.7.3. Adaptación a Java

```
import java.util.Scanner;

class Etapa {
    int horas;
    int minutos;
    int segundos;
}

public class TiempoCiclista {
    public static void main(String[] args) {
        // Crear un objeto Scanner para leer la entrada del usuario
        Scanner entrada = new Scanner(System.in);
        // Pedir al usuario que ingrese el número de etapas
        System.out.print("Ingrese el número de etapas:");
        int n = entrada.nextInt();
        // Crear un arreglo de etapas con tamaño n
        Etapa[] etapas = new Etapa[n];
        // Pedir al usuario que ingrese los tiempos de cada etapa
        for (int i = 0; i < n; i++) {
            System.out.println("Ingrese el tiempo para la etapa " + (i + 1) + " :");
            etapas[i] = leerTiempoEtapa(entrada);
        }
        // Calcular el tiempo total empleado en correr todas las etapas
        Etapa tiempoTotal = calcularTiempoTotal(etapas);
        // Mostrar el tiempo total empleado
        System.out.println("El tiempo total empleado es: " + tiempoTotal.horas + "
        horas, " +
```

```
tiempoTotal.minutos + " minutos," + tiempoTotal.segundos + " segundos");  
// Cerrar el objeto Scanner  
entrada.close();  
}  
  
// Función para leer el tiempo de una etapa desde la entrada del usuario  
public static Etapa leerTiempoEtapa(Scanner sc) {  
    Etapa etapa = new Etapa();  
    System.out.print("Horas:");  
    etapa.horas = sc.nextInt();  
    System.out.print("Minutos:");  
    etapa.minutos = sc.nextInt();  
    System.out.print("Segundos:");  
    etapa.segundos = sc.nextInt();  
    return etapa;  
}  
  
// Función para calcular el tiempo total empleado en correr todas las etapas  
public static Etapa calcularTiempoTotal(Etapa[] etapas) {  
    Etapa tiempoTotal = new Etapa();  
    // Inicializar los campos del tiempo total en 0  
    tiempoTotal.horas = 0;  
    tiempoTotal.minutos = 0;  
    tiempoTotal.segundos = 0;  
    // Sumar los tiempos de todas las etapas al tiempo total  
    for (Etapa etapa : etapas) {  
        tiempoTotal.horas += etapa.horas;  
        tiempoTotal.minutos += etapa.minutos;  
        tiempoTotal.segundos += etapa.segundos;  
    }  
}
```

```
// Ajustar los minutos y segundos si superan 60
tiempoTotal.minutos += tiempoTotal.segundos / 60;
tiempoTotal.segundos %= 60;
tiempoTotal.horas += tiempoTotal.minutos / 60;
tiempoTotal.minutos %= 60;
}
return tiempoTotal;
}
}
```


8.8.

Defina una estructura que sirva para representar a una persona. La estructura debe contener dos campos: el nombre de la persona y un valor de tipo lógico que indica si la persona tiene algún tipo de discapacidad. Realice un programa que dado un vector de personas rellene dos nuevos vectores: uno que contenga las personas que no tienen ninguna discapacidad y otro que contenga las personas con discapacidad.

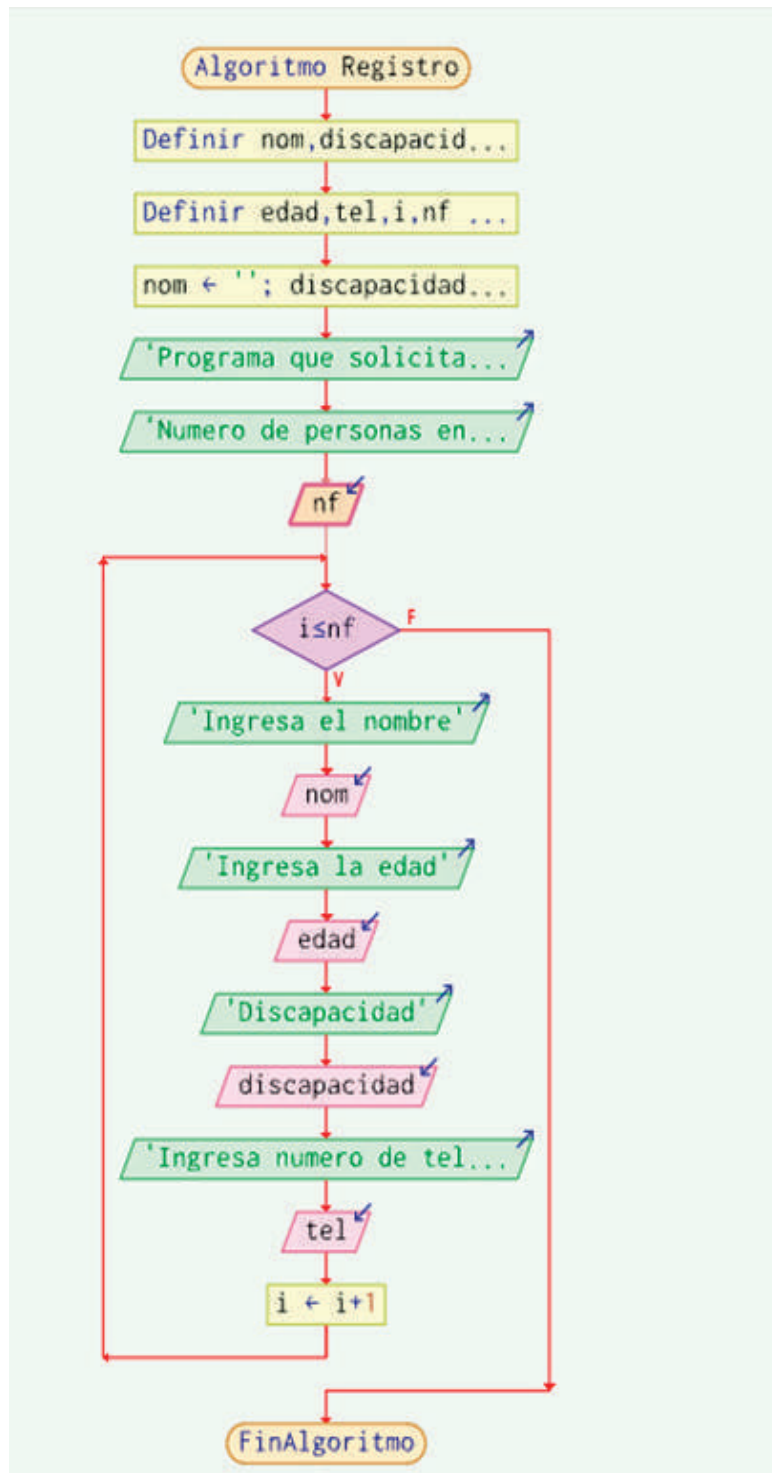
8.8.1. Código Pseint

Algoritmo Registro

```
Definir nom, discapacidad Como Caracter
Definir edad, tel, i, nf Como Entero
nom<-"";discapacidad<-"";nf<-0;edad<-0;tel<-0;i<-1;
Escribir "Programa que solicita datos de la persona";
Escribir "Numero de personas en la fila";
leer nf;
Mientras i<=nf Hacer
    Escribir "Ingresa el nombre";
    Leer nom;
    Escribir "Ingresa la edad";
    leer edad;
    Escribir "Discapacidad";
    leer discapacidad;
    Escribir "Ingresa numero de telefono";
    leer tel;
    i<-i+1;
FinMientras
```

FinAlgoritmo

8.8.2. Diagrama de flujo



8.8.3. Adaptación a Java

Para la clase “Persona”

```
package Representar;

import java.util.Scanner;
import java.util.ArrayList;

public class Persona {

    public static void llenarMatriz(String[] matriz){

        int opciones;

        Boolean condicion;

        ArrayList<String> conDisca = new ArrayList<>();
        ArrayList<String> sinDisca = new ArrayList<>();
        Scanner entrada = new Scanner(System.in);

        for (int i = 0; i<matriz.length;i++){
            System.out.println("Digite a la persona [" + i + "]:");
            matriz[i] = entrada.nextLine();
        }

        for (int i = 0; i<matriz.length;i++){

            System.out.println(matriz[i]+" presenta discapacidad: (0) NO (1) SI");

            opciones = entrada.nextInt();

            switch (opciones){

                case 0:
                    sinDisca.add(matriz[i]);
                    break;

                case 1:
                    conDisca.add(matriz[i]);
                    break;

            }

        }

        int longitud1 = conDisca.size();
        int longitud2 = sinDisca.size();
    }
}
```

```
System.out.println("Personas con discapacidad:");
for (int i = 0; i<longitud1; i++){
    String elemento = conDisca.get(i);
    System.out.println(elemento);
}
System.out.println("Personas sin discapacidad:");
for (int i = 0; i<longitud2; i++){
    String elementoDos = sinDisca.get(i);
    System.out.println(elementoDos);
}
}
}
```

Para la clase "Main"

```
package Representar;
import java.util.Scanner;
import java.util.Vector;
public class Main {
    public static void main(String[] args){
        Scanner entrada = new Scanner(System.in);
        String nombre;
        int eleccionUno, cantidad, disca, noDisca;
        Boolean discapacidad = false;
        Persona cadena = new Persona();
        System.out.println("Total de personas a registrar:");
        cantidad = entrada.nextInt();
        String general[] = new String[cantidad];
        Persona.llenarMatriz(general);
    }
}
```

Bloque 09: Funciones

9.1.

Escriba una función llamada mult() que acepte dos números en punto flotante como parámetros, multiplique estos dos números y despliegue el resultado.

9.1.1. Código Pseint

Funcion resultado <- mult(a,b)

resultado = a*b

Fin Funcion

Algoritmo B9_Ejercicio_1

Definir a,b, resultado Como Real

Escribir "Ingresar el valor de a:"

Leer a

Escribir "Ingresar el valor de b:"

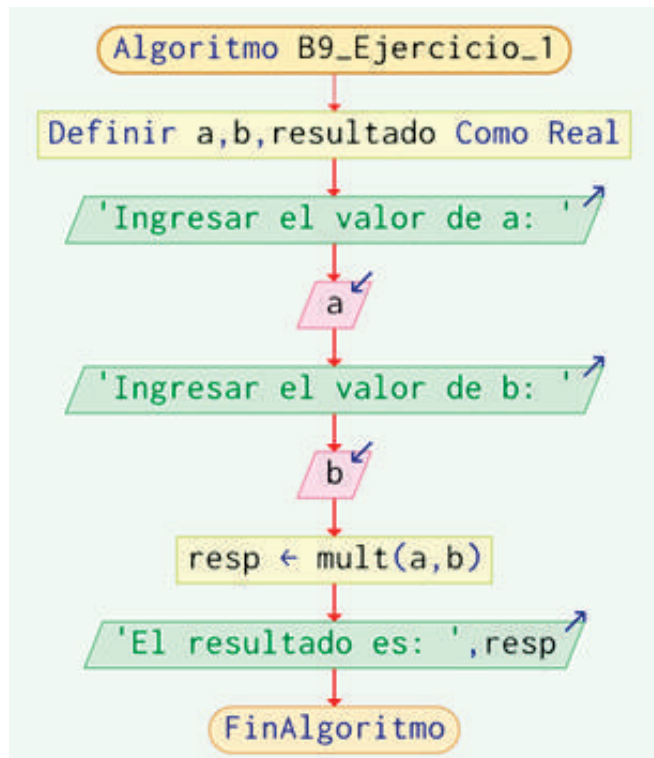
Leer b

resp = mult (a,b)

Escribir "El resultado es:", resp

FinAlgoritmo

9.1.2. Diagrama de flujo



9.1.3. Adaptación a Java

```
public class Funciones {  
    public static void main(String[] args) {  
        mult(4.5 , 7.5);  
    }  
    public static void mult(double num1, double num2) {  
        double resultado = num1 * num2;  
        System.out.println("El resultado de la multiplicación es:" + resultado);  
    }  
}
```

9.2.

Escriba una función llamada `al_cuadrado()` que calcule el cuadrado del valor que se le transmite y despliegue el resultado. La función deberá ser capaz de elevar al cuadrado números flotantes.

9.2.1. Código Pseint

Algoritmo CUADRADO

Definir n, cuadro Como Entero

Escribir "Ingrese un numero:"

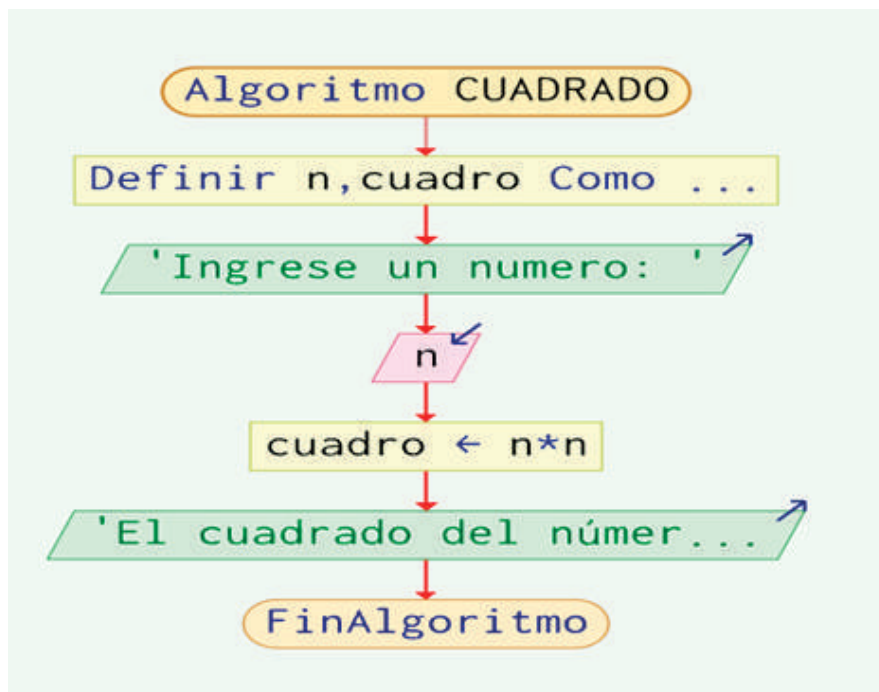
leer n

$\text{cuadro} = n * n$

Escribir "El cuadrado del número ",n," es:",cuadro

FinAlgoritmo

9.2.2. Diagrama de flujo



9.2.3. Adaptación a Java

```
public class Funciones {  
    public static void main(String[] args) {  
        al_cuadrado(5.5);  
    }  
    public static void al_cuadrado(double num) {  
        double resultado = num * num;  
        System.out.println("El cuadrado de " + num + " es: " + resultado);  
    }  
}
```

9.3.

Escriba una función nombrada `funpot()` que eleve un número entero que se le transmita a una potencia en número entero positivo y despliegue el resultado. El número entero positivo deberá ser el segundo valor transmitido a la función.

9.3.1. Código Pseint

Funcion resultado <- funpot(base, exponente)

Definir resultado como Entero

resultado <- 1

Para i <- 1 Hasta exponente Hacer

 resultado <- resultado * base

FinPara

FinFuncion

Algoritmo B9_Ejercicio_3

Definir numBase, numExponente como Entero

 Escribir "Ingrese la base:"

Leer numBase

Escribir "Ingrese el exponente:"

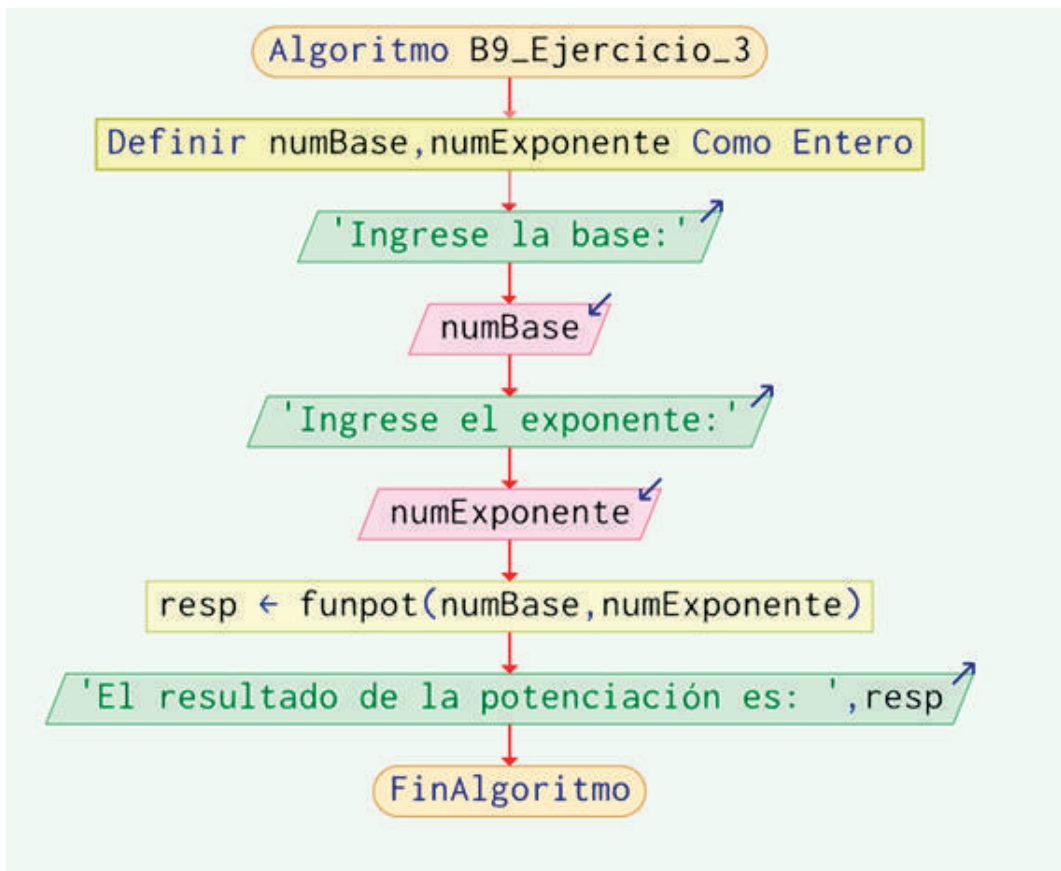
Leer numExponente

resp = funpot(numBase, numExponente)

Escribir "El resultado de la potenciación es:", resp

FinAlgoritmo

9.3.2. Diagrama de flujo



9.3.3. Adaptación a Java

```

public class Funciones {

    public static void main(String[] args) {
        funpot(5, 4);
    }
    public static void funpot(int base, int exponente) {
        int resultado = 1;
        for (int i = 0; i < exponente; i++) {
            resultado *= base;
        }
        System.out.println(base + " elevado a la potencia " + exponente + " es:" +
            resultado);
    }
}

```

94.

Escriba un programa en Java que devuelva la parte fraccionaria de cualquier número introducido por el usuario. Por ejemplo, si se introduce el número 256.879, debería desplegarse el número 0.879.

94.1. Código Pseint

Algoritmo devolverFraccion

Escribir "Por favor ingrese un número"

leer n

numerador<-1

denominador<-2

Para i<-1 Hasta n Hacer

si i%2=0 Entonces

 Escribir "-" numerador "/" denominador

SiNo

 Escribir "+" numerador "/" denominador

FinSi

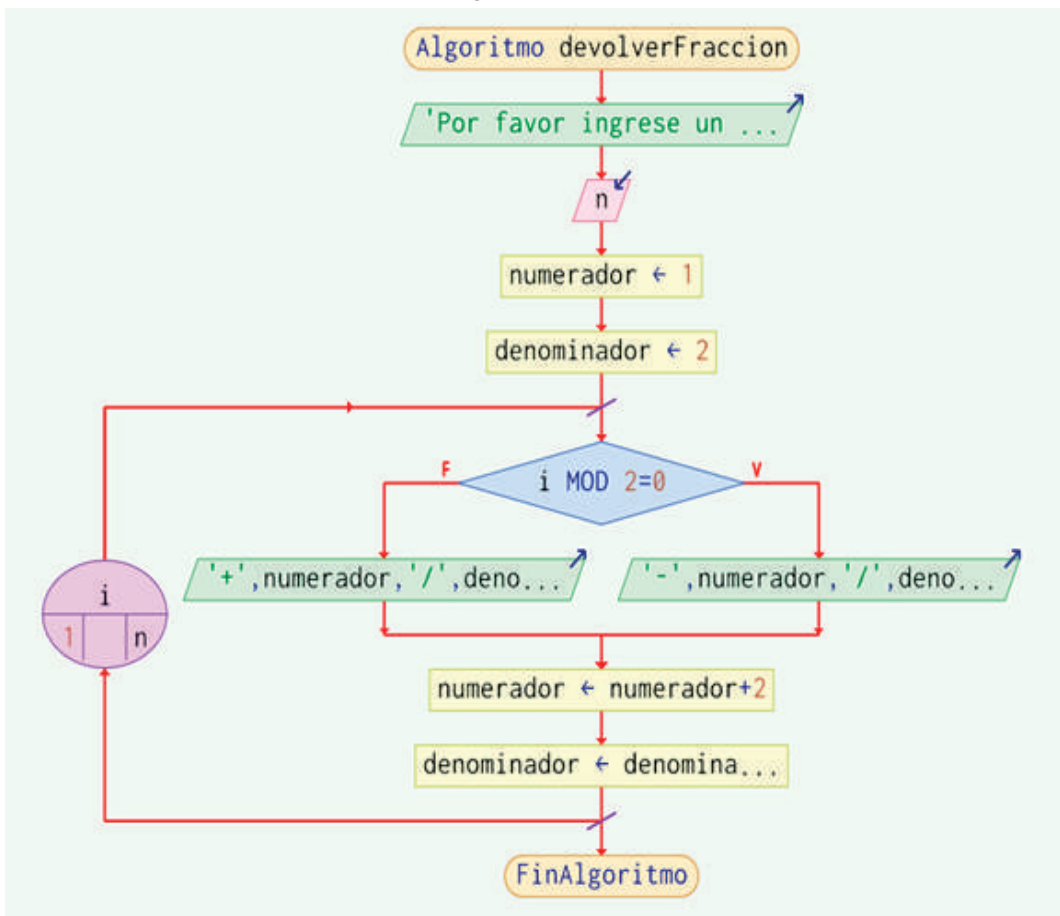
numerador<-numerador+2

denominador<-denominador+2

FinPara

FinAlgoritmo

94.2. Diagrama de flujo



94.3. Adaptación a Java

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

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

        Scanner scanner = new Scanner(System.in);
        System.out.print("Introduce un número:");

        double numero = scanner.nextDouble();
        double parteFraccionaria = numero % 1;

        System.out.println("La parte fraccionaria del número es: " + parteFraccionaria);
    }
}
```

9.5.

Escriba una plantilla de función llamada `despliegue()` que despliegue el valor del argumento único que se le transmite cuando es invocada la función.

9.5.1. Código Pseint

Funcion resultado <- despliegue(valor)

```
    resultado = valor
Fin Funcion
```

Algoritmo B9_Ejercicio_5

```
    Definir num como Entero
```

```
    Escribir "Ingrese un número:"
```

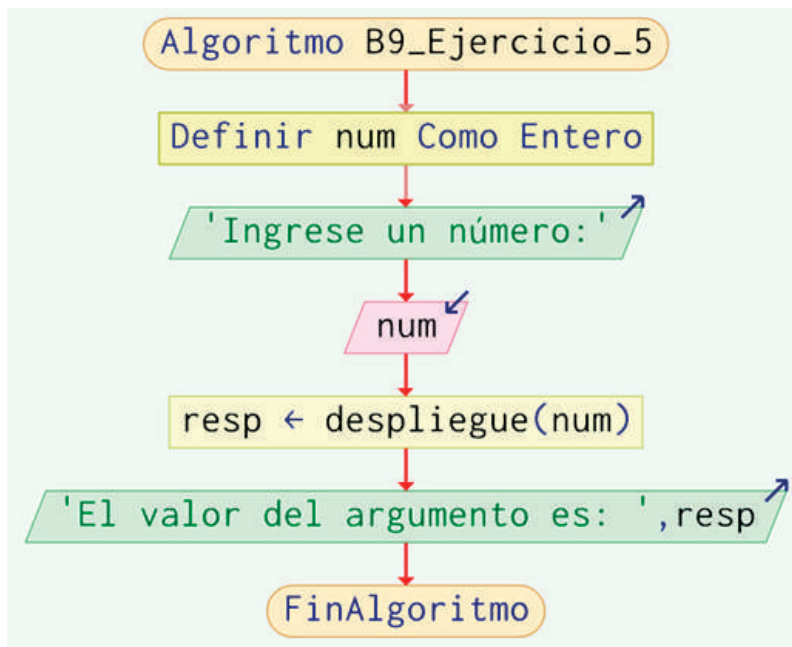
```
    Leer num
```

```
    resp = despliegue(num)
```

```
    Escribir "El valor del argumento es:", resp
```

FinAlgoritmo

9.5.2. Diagrama de flujo



9.5.3. Adaptación a Java

```
public class Funciones {  
    public static void main(String[] args) {  
        int numero = 23;  
        despliegue(numero);  
    }  
    public static void despliegue(int valor) {  
        System.out.println("El valor del argumento es:" + valor);  
    }  
}
```

9.6.

Escriba una plantilla de función llamada máximo () que devuelva el valor máximo de tres argumentos que se transmitan a la función cuando sea llamada. Suponga que los tres argumentos serán del mismo tipo de datos.

9.6.1. Código Pseint

Funcion rta <- mayor (xx,yy)

Definir rta Como Entero;

Si xx > yy Entonces

 rta <- xx;

SiNo

 Si yy > xx Entonces

 rta <- yy;

 SiNo

 rta <- -1

FinSi

FinSi
Fin Función

Algoritmo funciones

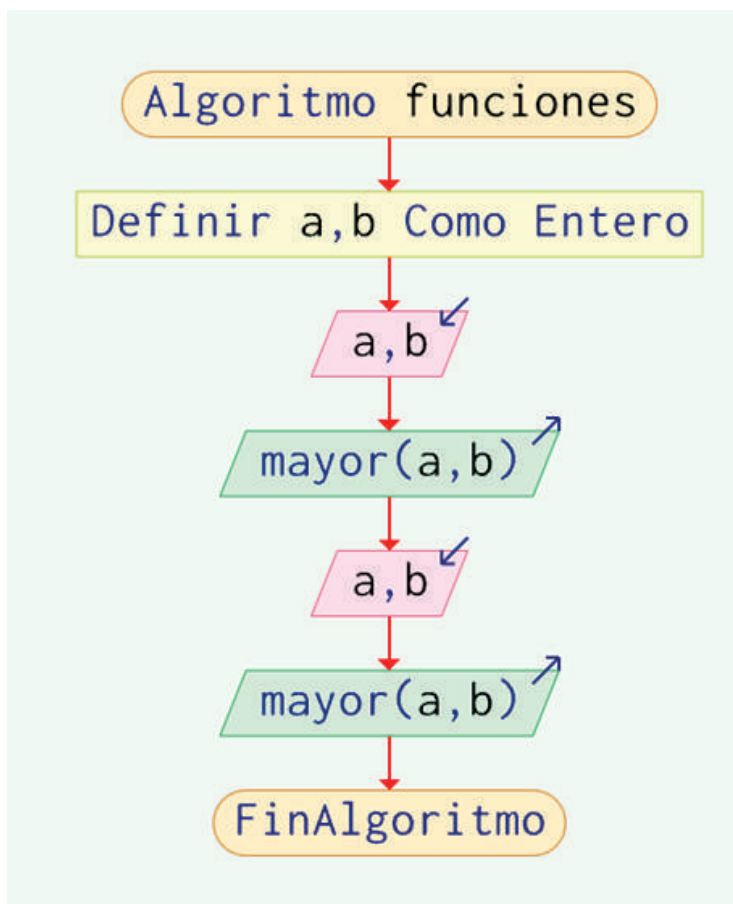
Definir a,b Como Entero;
leer a,b;

Escribir mayor(a,b);
leer a,b

Escribir mayor(a,b);

FinAlgoritmo

9.6.2. Diagrama de flujo



9.6.3. Adaptación a Java

```
public class Funciones {  
    public static <T extends Comparable<T>> T maximo(T a,T b,T c) {  
        T max = a;  
        if (b.compareTo(max) > 0) {  
            max = b;  
        }  
        if (c.compareTo(max) > 0) {  
            max = c;  
        }  
        return max;  
    }  
    public static void main(String[] args) {  
        Integer maximoEnteros = maximo(10, 5, 8);  
        System.out.println(maximoEnteros);  
        Double maximoDoubles = maximo(2.5, 4.7, 3.2);  
        System.out.println(maximoDoubles);  
        String maximoStrings = maximo("hola", "adiós", "saludos");  
        System.out.println(maximoStrings);  
    }  
}
```

9.7.

Intercambiar el valor de 2 variables utilizando paso de parámetros por referencia.

9.7.1 Código Pseint

Funcion intercambiarPorReferencia(a, b)

Definir temp Como Entero

temp <- a

a <- b

b <- temp

FinFuncion

Algoritmo B9_Ejercicio_7

Definir num1, num2 Como Entero

Escribir "Ingrese el valor de la variable num1:"

Leer num1

Escribir "Ingrese el valor de la variable num2:"

Leer num2

Escribir "Antes del intercambio:"

Escribir "num1 =", num1

Escribir "num2 =", num2

intercambiarPorReferencia(num1,num2)

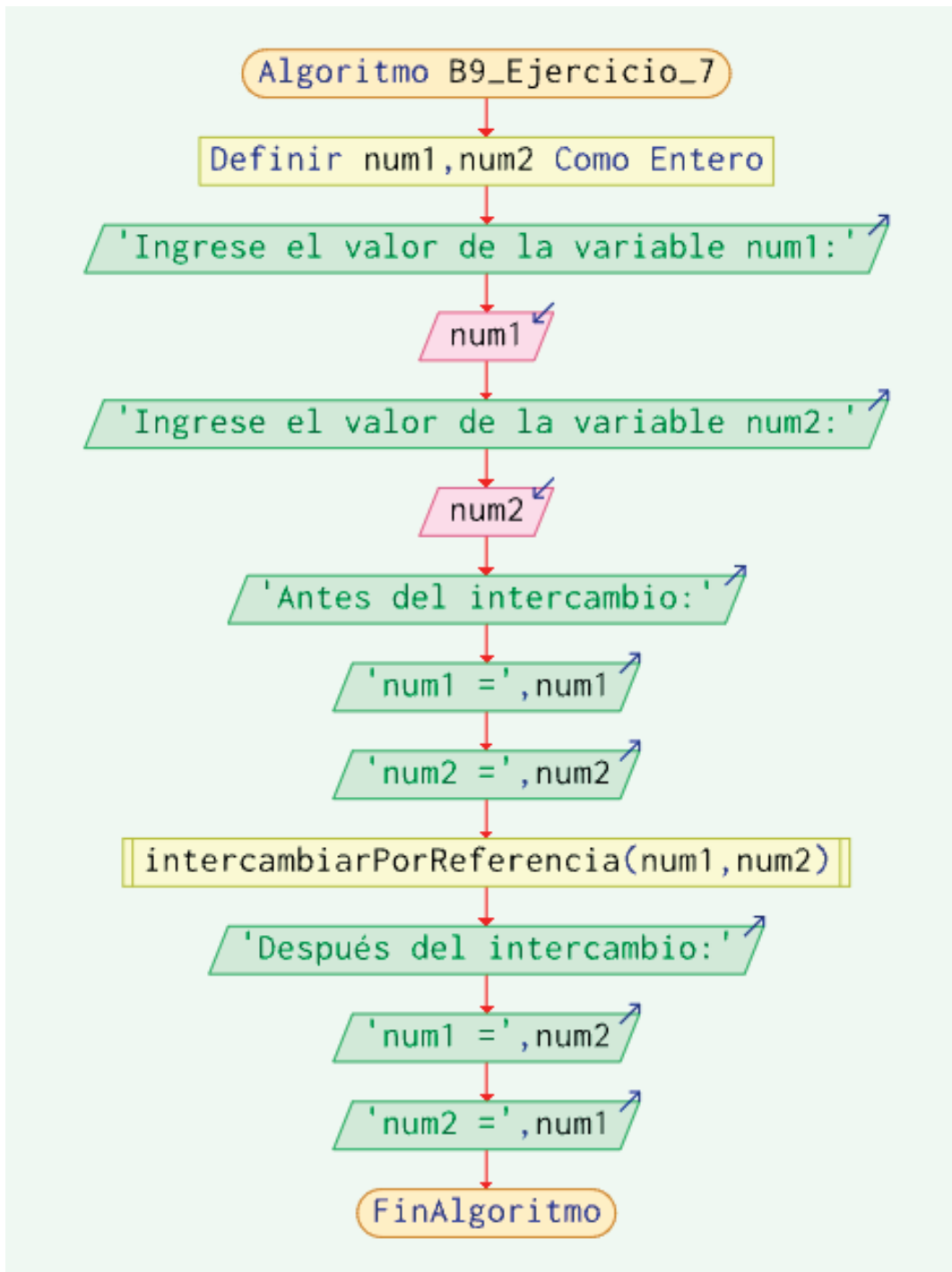
Escribir "Después del intercambio:"

Escribir "num1 =", num2

Escribir "num2 =", num1

FinAlgoritmo

9.7.2 Diagrama de flujo



9.7.3 Adaptación a Java

```
import java.util.Scanner;

public class Funciones {

    public static void main(String[] args) {

        Scanner teclado = new Scanner (System.in);

        int a ;
        int b;

        System.out.println("Ingrese la primera variable:");
        a =teclado.nextInt();

        System.out.println("Ingrese la primera variable:");
        b = teclado.nextInt();

        System.out.println("Antes del intercambio:");

        System.out.println("a = " + a);
        System.out.println("b = " + b);

        intercambiar(a, b);

        System.out.println("Despues del intercambio:");

        System.out.println("a = " + b);
        System.out.println("b = " + a);

    }

    public static void intercambiar(int x, int y) {

        int[] temp = new int[1];

        temp[0] = x;

        x = y;
        y = temp[0];

    }

}
```

9.8

Escriba una función nombrada cambio () que tenga un parámetro en número entero y seis parámetros de referencia en número entero nombrados cien, cincuenta, veinte, diez, cinco y uno, respectivamente. La función tiene que considerar el valor entero transmitido como una cantidad en dólares y convertir el valor en el número menor de billetes equivalentes.

9.8.1 Código Pseint

Algoritmo dolares

Escribir "Ingresar la cantidad en dolares:"

leer d

mon5<-trunc(d/5)

Res1<-d mod 5

mon2<-trunc(Res1/2)

mon1<-Res1 mod 2

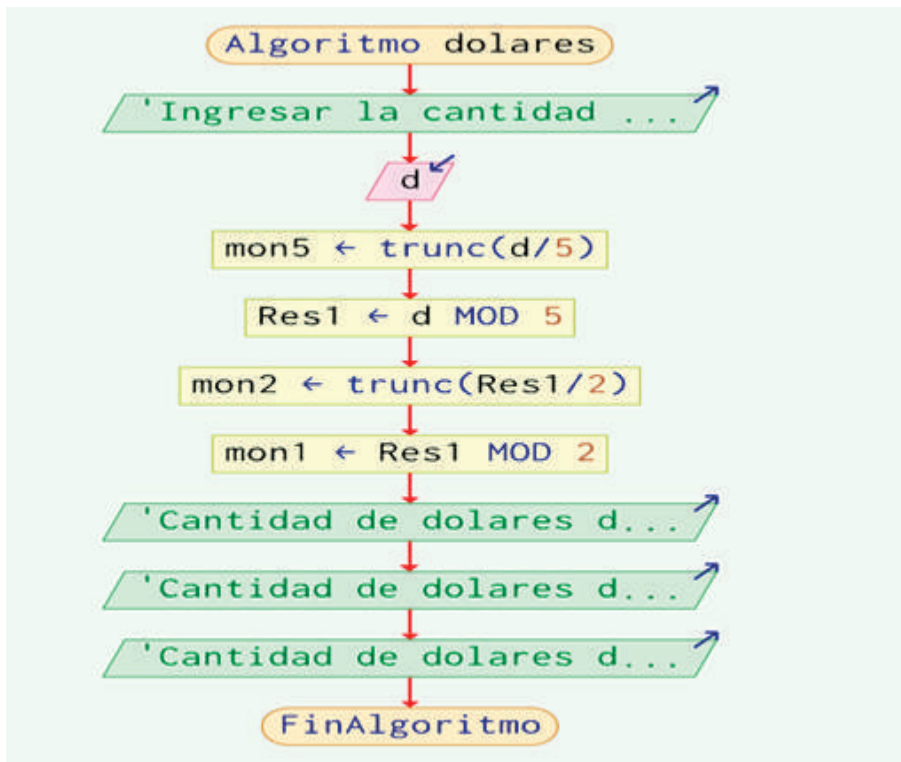
Escribir "Cantidad de dolares de 5:",mon5

Escribir "Cantidad de dolares de 2:",mon2

Escribir "Cantidad de dolares de 1:",mon1

FinAlgoritmo

9.8.2 Diagrama de flujo



9.8.3 Adaptación a Java

```

import java.util.Scanner;

public class Main {

    static int cien = 0, cincuenta = 0, veinte = 0, diez = 0, cinco = 0, uno = 0;

    public static void main(String[] args) {

        int cantidad;

        Scanner teclado = new Scanner(System.in);

        System.out.println("Ingrese la cantidad de billetes:");

        cantidad = teclado.nextInt();

        cambio(cantidad);
    }
}
  
```

```
System.out.println("Cantidad de billetes equivalentes para $" + cantidad + ":");
System.out.println("Billetes de $100:" + cien);
System.out.println("Billetes de $50:" + cincuenta);
System.out.println("Billetes de $20:" + veinte);
System.out.println("Billetes de $10:" + diez);
System.out.println("Billetes de $5:" + cinco);
System.out.println("Billetes de $1:" + uno);
}
public static void cambio(int cantidad) {
    cien = cantidad / 100;
    cantidad %= 100;
    cincuenta = cantidad / 50;
    cantidad %= 50;
    veinte = cantidad / 20;
    cantidad %= 20;
    diez = cantidad / 10;
    cantidad %= 10;
    cinco = cantidad / 5;
    cantidad %= 5;
    uno = cantidad;
}
}
```

9.9

Escriba una función nombrada tiempo() que tenga un parámetro en número entero llamado totalSeg y tres parámetros de referencia enteros nombrados horas, min y seg. La función es convertir el número de segundos transmitido en un número equivalente de horas, minutos y segundos.

9.9.1 Código Pseint

Funcion ConvertirTiempo(segundos)

Definir horas, minutos, seg Como Entero

horas <- TRUNC(segundos / 3600)

minutos <- TRUNC((segundos - (horas * 3600)) / 60)

seg <- segundos - (horas * 3600) - (minutos * 60)

Escribir "Horas : ", horas

Escribir "Minutos : ", minutos

Escribir "Segundos : ", seg

FinFuncion

Algoritmo B9_Ejercicio_9

Definir segundoss Como Entero

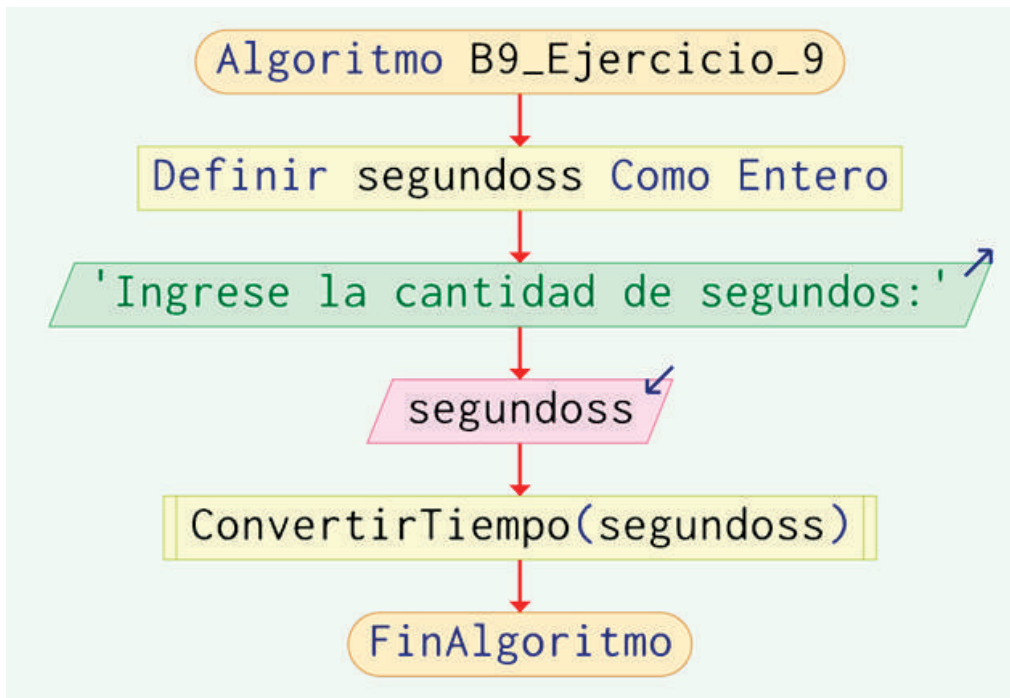
Escribir "Ingrese la cantidad de segundos:"

Leer segundoss

ConvertirTiempo(segundoss)

FinAlgoritmo

9.9.2 Diagrama de flujo



9.9.3 Adaptación a Java

```
public class Funciones {  
    public static void main(String[] args) {  
        int totalSegundos = 2400; //inserte los segundos aquí  
        Tiempo tiempo = calcularTiempo(totalSegundos);  
        System.out.println("Horas:" + tiempo.horas);  
        System.out.println("Minutos:" + tiempo.minutos);  
        System.out.println("Segundos:" + tiempo.segundos);  
    }  
    public static Tiempo calcularTiempo(int totalSeg) {  
        int horas = totalSeg / 3600;  
        int segundosRestantes = totalSeg % 3600;
```

```
        int minutos = segundosRestantes / 60;
        int segundos = segundosRestantes % 60;
        return new Tiempo(horas, minutos, segundos);
    }
}
class Tiempo {
    int horas;
    int minutos;
    int segundos;
    public Tiempo(int horas, int minutos, int segundos) {
        this.horas = horas;
        this.minutos = minutos;
        this.segundos = segundos;
    }
}
```


9.10

Escriba una función nombrada `calc_años()` que tenga un parámetro entero que represente el número total de días desde la fecha 1/1/2000 y parámetros de referencia nombrados año, mes y día. La función es calcular el año, mes y día actual para el número dado de días que se le transmitan. Usando las referencias, la función deberá alterar en forma directa los argumentos respectivos en la función que llama. Para este problema suponga que cada año tiene 365 días y cada mes tiene 30 días.

9.10.1 Código Pseint

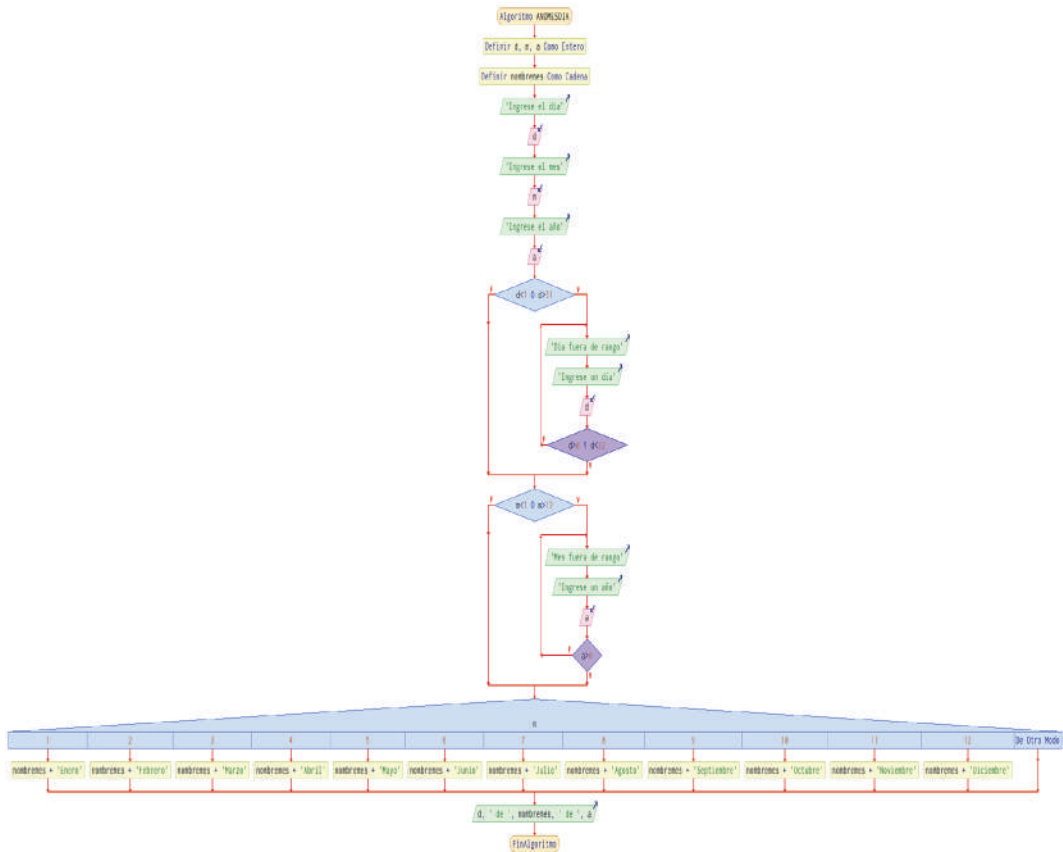
Algoritmo AñoMesDia

```
definir d,m,a Como Entero
Definir nombremes Como Caracter
Escribir "Ingrese el día"
leer d
Escribir "Ingrese el mes"
leer m
Escribir "Ingrese el año"
Leer a
si d<1 o d>31
  Repetir
    Escribir "Dia fuera de rango"
    Escribir "Ingrese un día"
    Leer d
  Hasta Que d>0 y d<32
FinSi
si m<1 o m>12
  Repetir
```

```
    Escribir "Mes fuera de rango"
    Escribir "Ingrese un año"
    leer a
    Hasta Que a>0
    FinSi
Segun m Hacer
    caso 1:nombremes="Enero"
    caso 2:nombremes="Febrero"
    caso 3:nombremes="Marzo"
    caso 4:nombremes="Abril"
    caso 5:nombremes="Mayo"
    caso 6:nombremes="Junio"
    caso 7:nombremes="Julio"
    caso 8:nombremes="Agosto"
    caso 9:nombremes="Septiembre"
    caso 10:nombremes="Octubre"
    caso 11:nombremes="Noviembre"
    caso 12:nombremes="Diciembre"
    FinSegun
    Escribir d," de ",nombremes," de ",a
```

FinAlgoritmo

9.10.2 Diagrama de flujo



9.10.3 Adaptación a Java

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

```
public class Funciones {
```

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

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

```
        System.out.print("Ingrese el número total de días desde 1/1/2000:");
```

```
        int totalDias = scanner.nextInt();
```

```
        int year, month, day;
```

```
        calc_años(totalDias, year = 2000, month = 1, day = 1);
```

```
    }
```

```
public static void calc_años(int totalDias, int year, int month, int day) {
    while (totalDias >= 365) {
        if (totalDias >= 366) {
            totalDias -= 366;
            year++;
        } else {
            totalDias -= 365;
            year++;
        }
    }
    while (totalDias >= 30) {
        totalDias -= 30;
        month++;
    }
    day += totalDias;
    if (day > 30) {
        month++;
        day -= 30;
    }
    if (month > 12) {
        year += month / 12;
        month %= 12;
    }
    System.out.println("El año actual es:" + day + "/" + month + "/" + year);
}
```

9.11

Realice una función que tome como parámetros un vector de números enteros y devuelva la suma de sus elementos.

9.11.1 Código Pseint

Funcion CalcularSumaVector(vector, dim)

Definir suma Como Real

suma <- 0

Para x <- 1 Hasta dim Con Paso 1 Hacer

 suma <- suma + vector[x]

FinPara

Escribir "La suma de los elementos es:", suma

FinFuncion

Algoritmo B9_Ejercicio_11

Definir x Como Entero

Definir suma, vector Como Real

Escribir "Ingresar dimensión:"

Leer dim

Dimension vector[dim]

Para x <- 1 Hasta dim Con Paso 1 Hacer

 Escribir "Ingresa el numero ", x, ":"

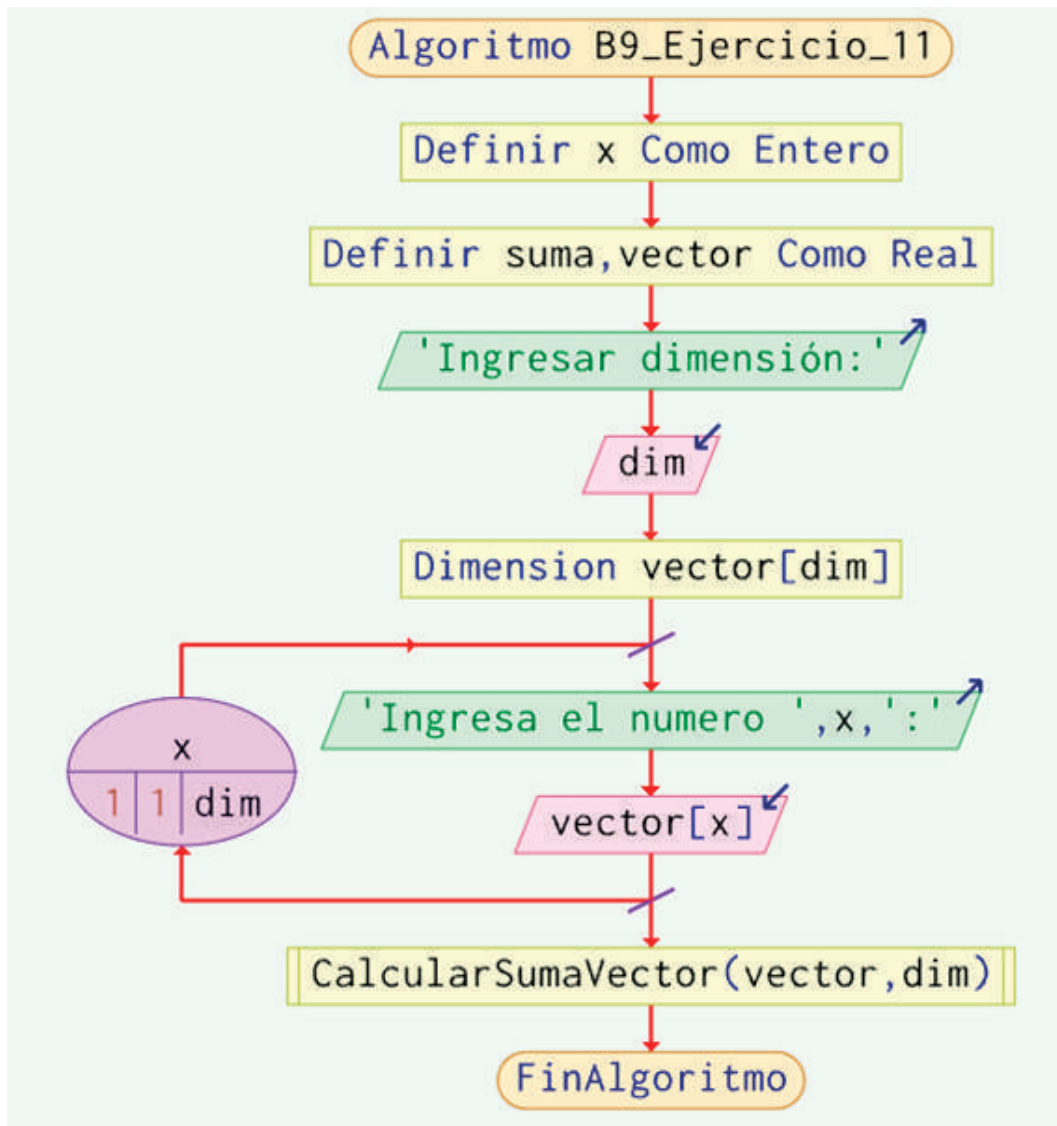
 Leer vector[x]

FinPara

CalcularSumaVector(vector, dim)

FinAlgoritmo

9.11.2 Diagrama de flujo



9.11.3 Adaptación a Java

```
public class Funciones {  
    public static int sumarElementos(int[] vector) {  
        int suma = 0;  
        for (int i = 0; i < vector.length; i++) {  
            suma += vector[i];  
        }  
        return suma;  
    }  
    public static void main(String[] args) {  
        int[] vector = {2, 4, 6, 8, 10};  
        int resultado = sumarElementos(vector);  
        System.out.println("La suma de los elementos del vector es:" + resultado);  
    }  
}
```

9.12

Realice una función que tome como parámetros un vector y su tamaño y diga si el vector está ordenado crecientemente. Sugerencia: compruebe que para todas las posiciones del vector, salvo para la 0, el elemento del vector es mayor o igual que el elemento que le precede en el vector

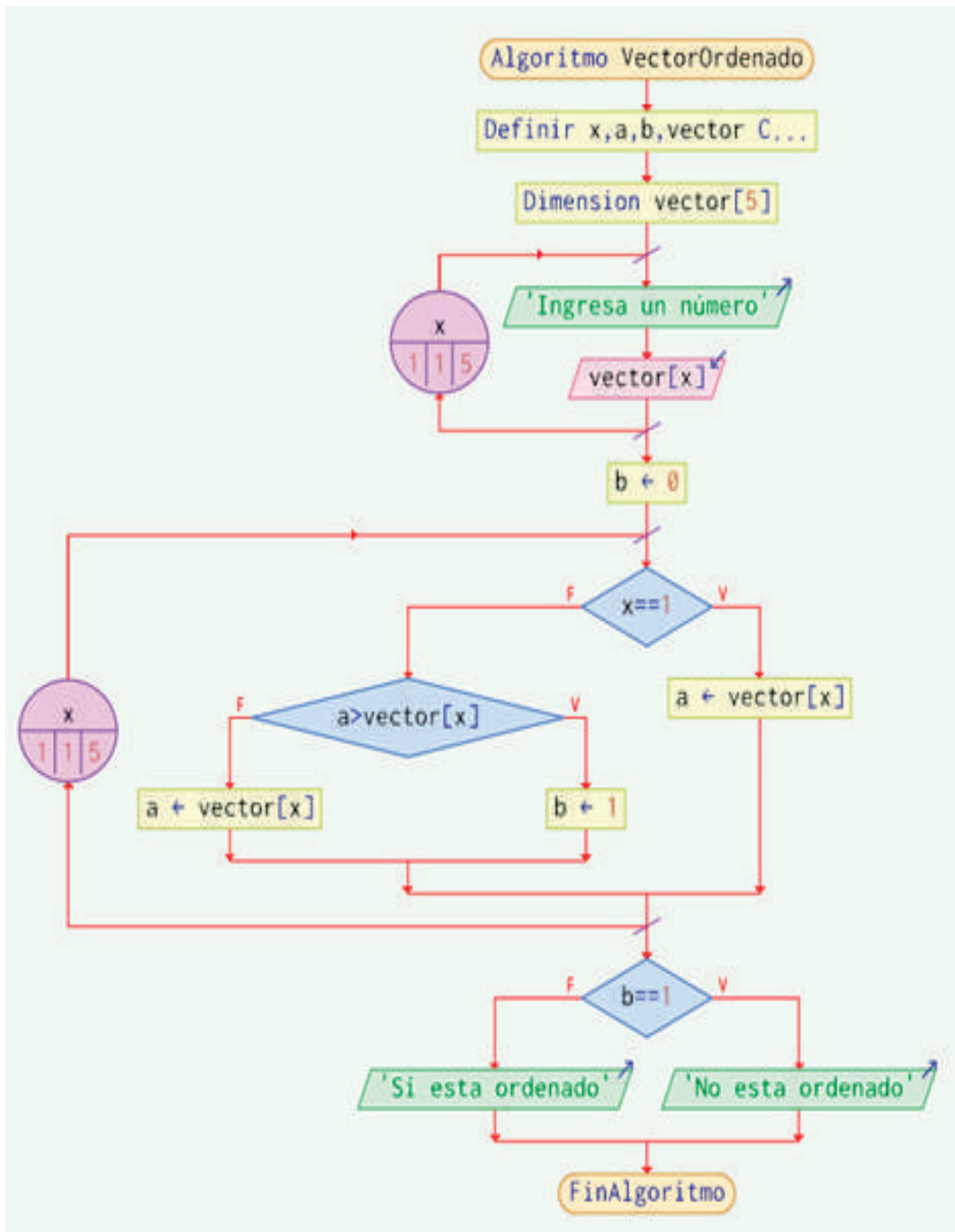
Código de Pseint

Algoritmo VectorOrdenado

```
Definir x,a,b,vector Como Entero
Dimension vector[5]
Para x = 1 Hasta 5 Con Paso 1 Hacer
    Escribir "Ingresa un número"
    leer vector(x)
FinPara
b=0
Para x = 1 Hasta 5 Con Paso 1 Hacer
    si x == 1 Entonces
        a = vector(x)
    SiNo
        si a > vector(x) Entonces
            b = 1
        SiNo
            a = vector(x)
    FinSi
FinSi
FinPara
si b == 1 Entonces
    Escribir "No esta ordenado"
SiNo
    Escribir "Si esta ordenado"
FinSi
```

FinAlgoritmo

Diagrama de flujo



Adaptación a JAVA:

```
public class Main {  
    public static boolean estaOrdenadoCrecientemente(int[] vector, int tamaño) {  
        for (int i = 1; i < tamaño; i++) {  
            if (vector[i] < vector[i - 1]) {  
                return false;  
            }  
        }  
        return true;  
    }  
    public static void main(String[] args) {  
        int[] vector1 = {1, 2, 3, 4, 5};  
        int tamaño1 = vector1.length;  
        System.out.println("¿El vector está ordenado crecientemente?" + estaOrdenadoCrecientemente(vector1, tamaño1));  
        int[] vector2 = {5, 4, 3, 2, 1};  
        int tamaño2 = vector2.length;  
        System.out.println("¿El vector está ordenado crecientemente?" + estaOrdenadoCrecientemente(vector2, tamaño2));  
    }  
}
```

Ejercicio 13:

Realiza una función que tome como parámetros un vector de números y su tamaño y cambie el signo de los elementos del vector.

Código de Pseudocódigo

Funcion CambiarSignoVector(vector, tamaño)

```
Para i <- 1 Hasta tamaño Hacer
    vector[i] <- -vector[i]
FinPara
```

FinFuncion

Algoritmo B9_Ejercicio_13

```
Definir tamaño Como Entero
Escribir "Ingrese el tamaño del vector:"
```

Leer tamaño

Dimension vector[tamaño]

Escribir "Ingrese los elementos del vector:"

```
Para i <- 1 Hasta tamaño Hacer
    Escribir "Ingrese el elemento ", i, ":"
```

Leer vector[i]

FinPara

Escribir "Vector original:"

```
Para i <- 1 Hasta tamaño Hacer
    Escribir vector[i]
```

FinPara

CambiarSignoVector(vector, tamaño)

Escribir "Vector con el signo cambiado:"

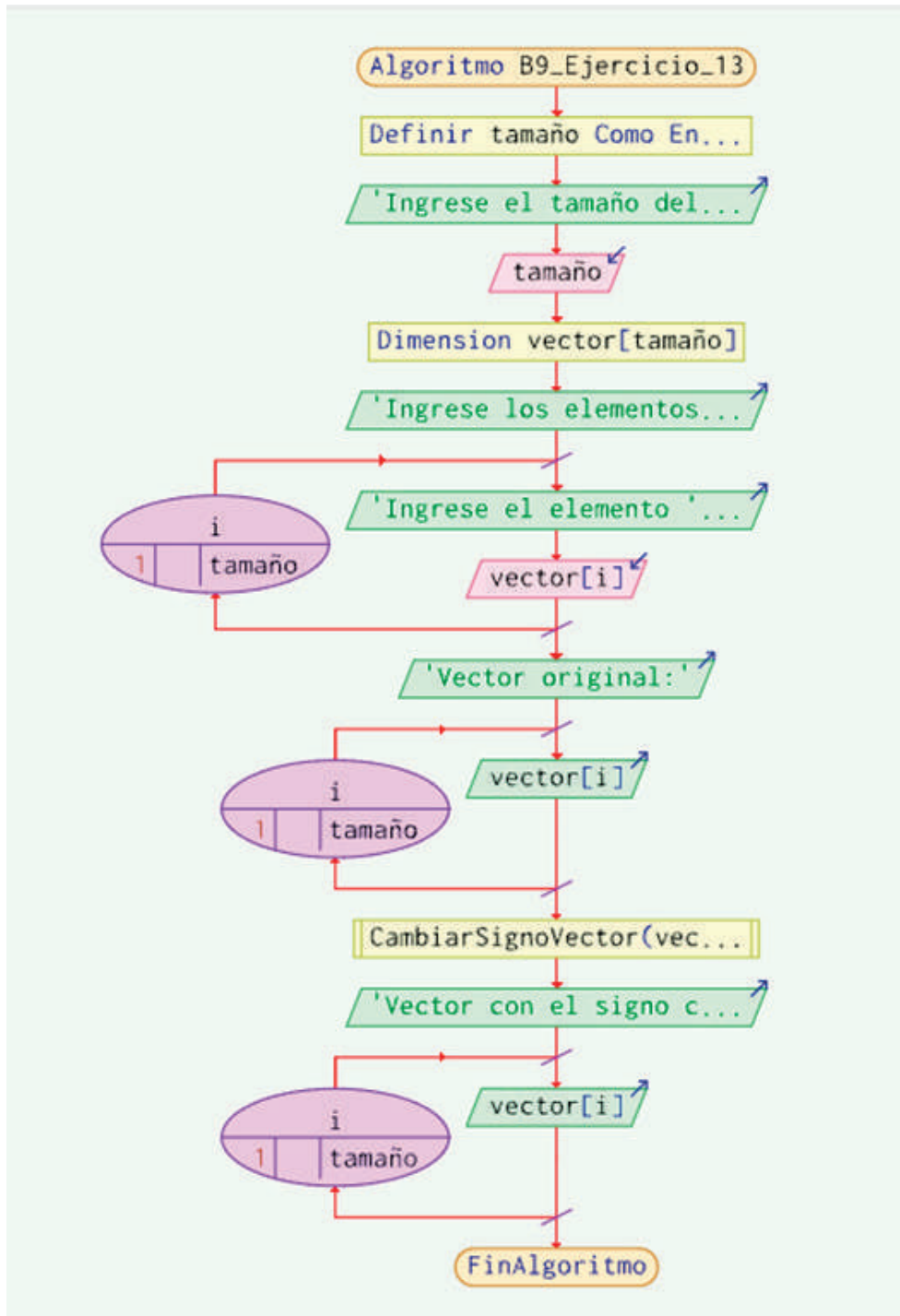
```
Para i <- 1 Hasta tamaño Hacer
```

Escribir vector[i]

FinPara

FinAlgoritmo

Diagrama de flujo



Adaptación a JAVA:

```
public class Main {  
    public static void cambiarSigno(int[] vector) {  
        for (int i = 0; i < vector.length; i++) {  
            vector[i] = -vector[i];  
        }  
    }  
  
    public static void main(String[] args) {  
        int[] vector = {1, -2, 3, -4, 5};  
        System.out.println("Vector original:");  
        imprimirVector(vector);  
        cambiarSigno(vector);  
        System.out.println("Vector con cambio de signo:");  
        imprimirVector(vector);  
    }  
  
    public static void imprimirVector(int[] vector) {  
        for (int i = 0; i < vector.length; i++) {  
            System.out.print(vector[i] + " ");  
        }  
        System.out.println();  
    }  
}
```

Ejercicio 14:

Realiza una función que tome como parámetros un vector de enteros y su tamaño e imprima un vector con los elementos impares del vector recibido.

Código Pseint

Algoritmo imprimirVectores

Definir x,a,f,c,n,m,matriz,vector Como Entero

Escribir "Ingrese las filas "

leer n

Escribir "Ingresa las columnas "

leer m

$x = n * m$

Dimension matriz[n,m],vector[x]

para f = 1 Hasta n Con Paso 1 Hacer

 para c = 1 Hasta m Con Paso 1 Hacer

 matriz(f,c) = azar(9) + 1

 FinPara

FinPara

a = 1

Para f = 1 Hasta n Con Paso 1 Hacer

 Para c = 1 Hasta m Con Paso 1 Hacer

 Escribir matriz(f,c)," " Sin Saltar
 vector(a) = matriz(f,c)

 a = a + 1

 FinPara

Escribir ""

FinPara

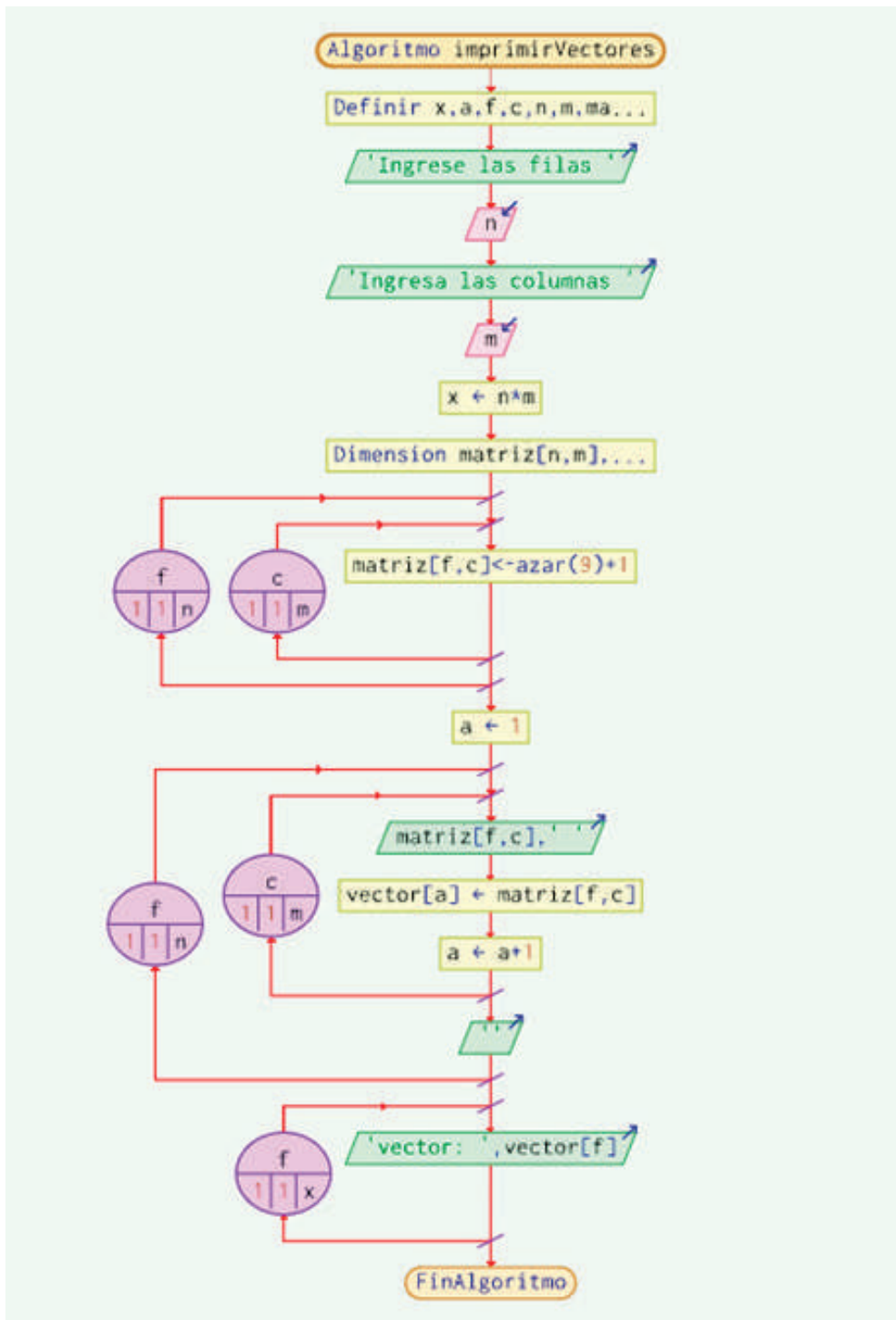
Para f = 1 Hasta x Con Paso 1 Hacer

 Escribir "vector:",vector(f)

FinPara

FinAlgoritmo

Diagrama de flujo



Adaptación a JAVA:

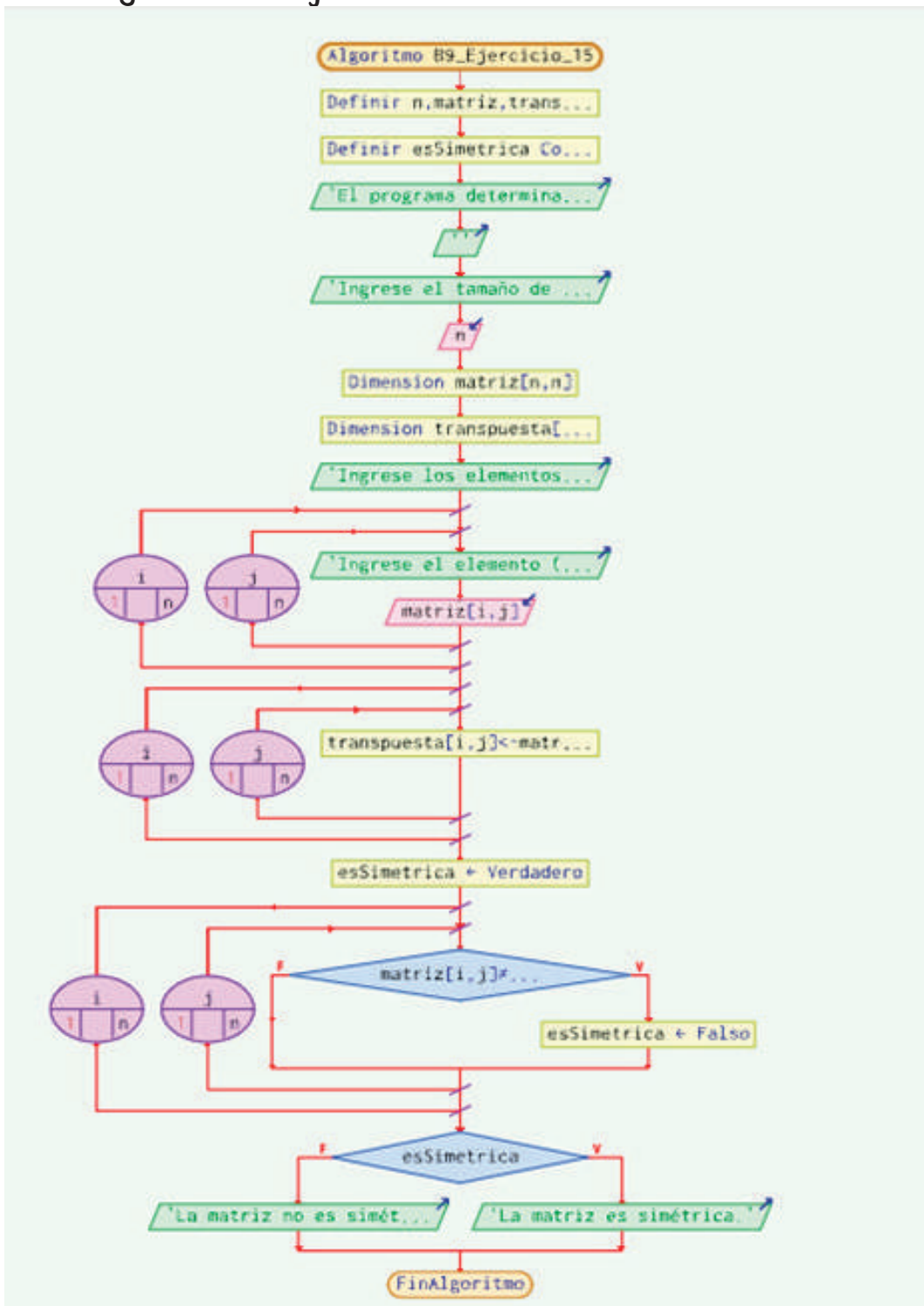
```
public class Main {  
    public static void cambiarSignoVector(int[] vector, int tamaño) {  
        for (int i = 0; i < tamaño; i++) {  
            vector[i] = -vector[i];  
        }  
    }  
    public static void main(String[] args) {  
        int[] miVector = {1, 2, 3, 4, 5};  
        int tamañoVector = miVector.length;  
        cambiarSignoVector(miVector, tamañoVector);  
        for (int i = 0; i < tamañoVector; i++) {  
            System.out.print(miVector[i] + " ");  
        }  
    }  
}
```



```
esSimetrica <- Verdadero
Para i <- 1 Hasta n Hacer
  Para j <- 1 Hasta n Hacer
    Si matriz[i, j] <> transpuesta[i, j] Entonces
      esSimetrica <- Falso
    FinSi
  FinPara
FinPara
Si esSimetrica Entonces
  Escribir "La matriz es simétrica."
Sino
  Escribir "La matriz no es simétrica."
FinSi
```

FinAlgoritmo

Diagrama de flujo



Adaptación a JAVA:

```
public class Main{  
    public static void imprimirImpares(int[] vector, int tamaño) {  
        int contadorImpares = 0;  
        for (int i = 0; i < tamaño; i++) {  
            if (vector[i] % 2 != 0) {  
                contadorImpares++;  
            }  
        }  
        int[] vectorImpares = new int[contadorImpares];  
        int indicelmpares = 0;  
        for (int i = 0; i < tamaño; i++) {  
            if (vector[i] % 2 != 0) {  
                vectorImpares[indicelmpares] = vector[i];  
                indicelmpares++;  
            }  
        }  
        for (int i = 0; i < contadorImpares; i++) {  
            System.out.print(vectorImpares[i] + " ");  
        }  
    }  
    public static void main(String[] args) {  
        int[] miVector = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
        int tamañoVector = miVector.length;  
        imprimirImpares(miVector, tamañoVector);  
    }  
}
```

Ejercicio 16:

Realice una función que dada una matriz y un número de fila de la matriz devuelva el menor de los elementos almacenados en dicha fila.

Código Pseint

Algoritmo MatrizMenor

Definir n,x,mayorI,menorI,vector Como Entero

Escribir "Ingresa el tamaño del vector"

leer n

Dimension vector[n]

para x = 1 Hasta n Con Paso 1 Hacer

Escribir "Ingresa un numero"

Leer vector(x)

FinPara

mayorI = 0

menorI = 0

Para x = 1 Hasta n Con Paso 1 Hacer

Escribir vector(x)

si x == 1 Entonces

menorI = vector(x)

mayorI = vector(x)

SiNo

si vector(x) > mayorI Entonces

mayorI = vector(x)

SiNo

si vector(x) < menorI Entonces

menorI = vector(x)

FinSi

FinSi

FinSi

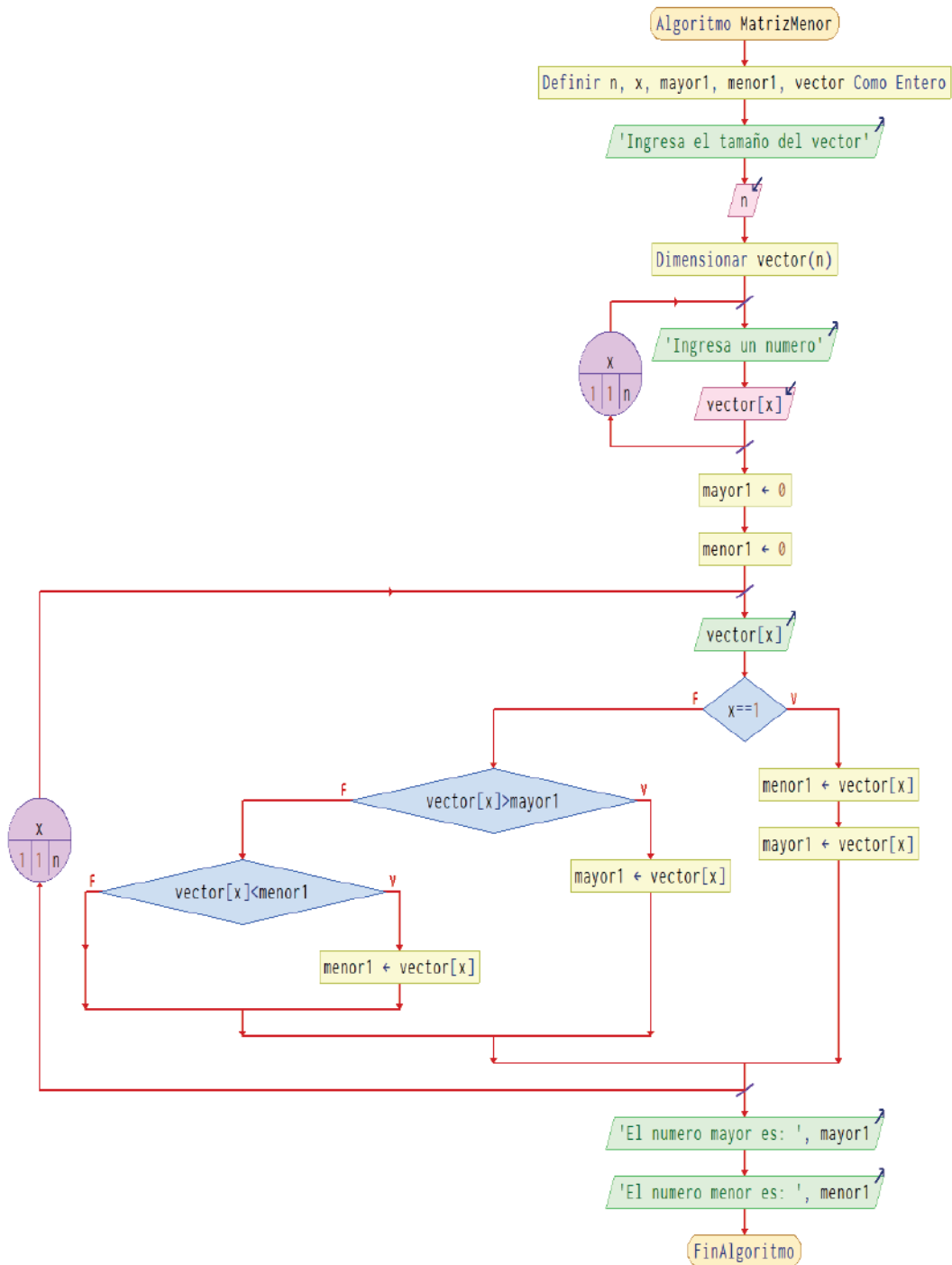
FinPara

Escribir "El numero mayor es:",mayorI

Escribir "El numero menor es:",menorI

FinAlgoritmo

Diagrama de flujo



Adaptación a JAVA:

```
public class Main {  
    public static boolean esSimetrica(int[][] matriz) {  
        int filas = matriz.length;  
        int columnas = matriz[0].length;  
        if (filas != columnas) {  
            return false;  
        }  
        for (int i = 0; i < filas; i++) {  
            for (int j = 0; j < columnas; j++) {  
                if (matriz[i][j] != matriz[j][i]) {  
                    return false;  
                }  
            }  
        }  
        return true;  
    }  
    public static void main(String[] args) {  
        int[][] matrizSimetrica = {  
            {1, 2, 3},  
            {2, 4, 5},  
            {3, 5, 6}  
        };  
        int[][] matrizNoSimetrica = {  
            {1, 2, 3},  
            {4, 5, 6},  
            {7, 8, 9}  
        };  
    }  
}
```

```
if (esSimetrica(matrizSimetrica)) {  
    System.out.println("La matriz es simétrica.");  
    System.out.println("La matriz es simétrica.");  
} else {  
    System.out.println("La matriz no es simétrica.");  
}  
if (esSimetrica(matrizNoSimetrica)) {  
    System.out.println("La matriz es simétrica.");  
} else {  
    System.out.println("La matriz no es simétrica.");  
}  
}  
}
```


Ejercicio 17:

Realice una función que dada una matriz y un número de fila de la matriz devuelva el menor de los elementos almacenados en dicha fila

Código Pseint

Funcion SumaNumerosComplejos(real1, imaginaria1, real2, imaginaria2)

```
Definir suma_real, suma_imaginaria Como Real
suma_real <- real1 + real2
suma_imaginaria <- imaginaria1 + imaginaria2
```

```
Escribir "El resultado de la suma es:"
```

```
Escribir "Parte Real:", suma_real
```

```
Escribir "Parte Imaginaria:", suma_imaginaria
```

```
FinFuncion
```

```
Algoritmo B9_Ejercicio_17
```

```
Definir real1, imaginaria1, real2, imaginaria2 Como Real
```

```
Escribir "Ingrese el primer número complejo:"
```

```
Escribir "Parte Real:"
```

```
Leer real1
```

```
Escribir "Parte Imaginaria:"
```

```
Leer imaginaria1
```

```
Escribir "Ingrese el segundo número complejo:"
```

```
Escribir "Parte Real:"
```

```
Leer real2
```

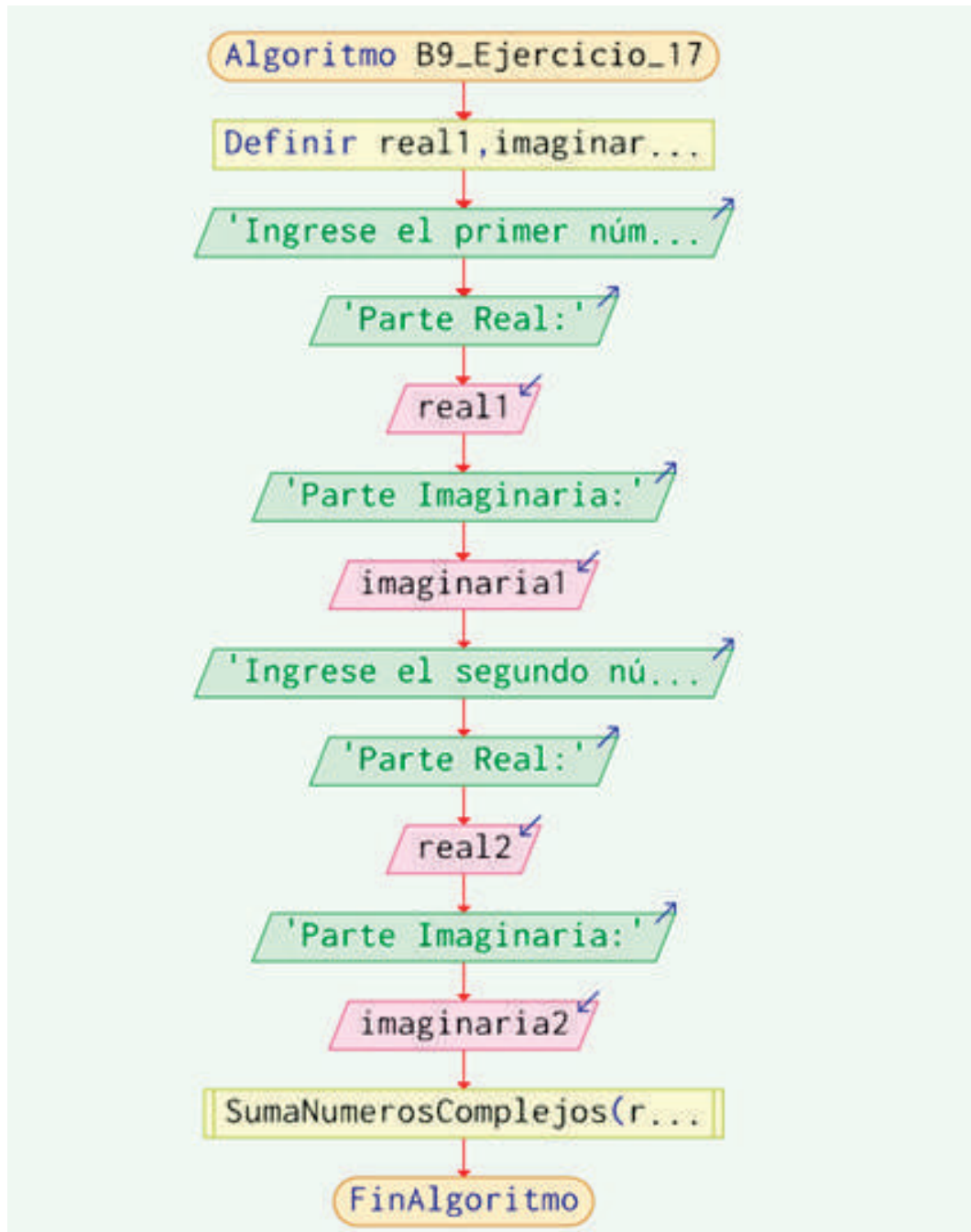
```
Escribir "Parte Imaginaria:"
```

```
Leer imaginaria2
```

```
SumaNumerosComplejos(real1, imaginaria1, real2, imaginaria2)
```

```
FinAlgoritmo
```

Diagrama de flujo



Adaptación a JAVA:

```
public class Main {  
    public static int obtenerMenorElemento(int[][] matriz, int numeroFila) {  
        int menorElemento = matriz[numeroFila][0];  
        for (int j = 1; j < matriz[numeroFila].length; j++) {  
            if (matriz[numeroFila][j] < menorElemento) {  
                menorElemento = matriz[numeroFila][j];  
            }  
        }  
        return menorElemento;  
    }  
    public static void main(String[] args) {  
        int[][] matriz = {  
            {5, 8, 3},  
            {2, 4, 9},  
            {7, 1, 6}  
        };  
        int numeroFila = 1;  
        int menorElemento = obtenerMenorElemento(matriz, numeroFila);  
        System.out.println("El menor elemento de la fila " + numeroFila + " es: " +  
            menorElemento);  
    }  
}
```

9.17

Suma de números Complejos: La suma de números complejos se realiza de la siguiente manera:

Un número complejo está compuesto por dos partes: una parte real y una parte imaginaria. La parte real se representa como “a” y la parte imaginaria como “b”. Si tenemos dos números complejos, el primero sería “a1 + b1i” y el segundo sería “a2 + b2i”, la suma de estos dos números complejos se obtiene sumando las partes reales por separado y las partes imaginarias por separado.

La fórmula para sumar dos números complejos es la siguiente:

$$(a_1 + b_1i) + (a_2 + b_2i) = (a_1 + a_2) + (b_1 + b_2)i$$

Por lo tanto, para sumar dos números complejos, simplemente sumamos sus partes reales y sus partes imaginarias por separado.

9.17.1 Código Pseint

Funcion SumaNumerosComplejos(real1, imaginaria1, real2, imaginaria2)

Definir suma_real, suma_imaginaria Como Real

suma_real <- real1 + real2

suma_imaginaria <- imaginaria1 + imaginaria2

Escribir “El resultado de la suma es:”

Escribir “Parte Real:”, suma_real

Escribir “Parte Imaginaria:”, suma_imaginaria

FinFuncion

Algoritmo B9_Ejercicio_17

Definir real1, imaginaria1, real2, imaginaria2 Como Real

Escribir “Ingrese el primer número complejo:”

Escribir “Parte Real:”

Leer real1

Escribir “Parte Imaginaria:”

Leer imaginaria1

Escribir "Ingrese el segundo número complejo:"

Escribir "Parte Real:"

Leer real2

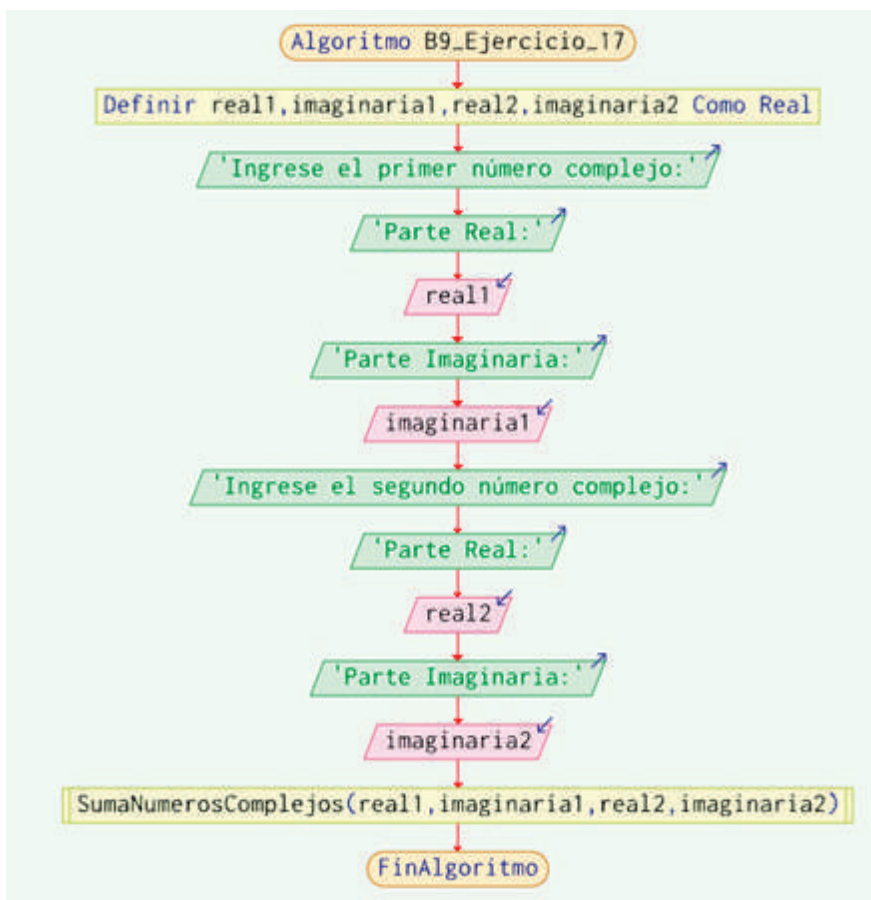
Escribir "Parte Imaginaria:"

Leer imaginaria2

SumaNumerosComplejos(real1, imaginaria1, real2, imaginaria2)

FinAlgoritmo

9.17.2 Diagrama de flujo



9.17.3 Adaptación JAVA

```
public class Main {
    private double parteReal;
    private double partelmaginaria;
    public NumeroComplejo(double parteReal, double partelmaginaria) {
        this.parteReal = parteReal;
        this.partelmaginaria = partelmaginaria;
    }
    public static NumeroComplejo sumaComplejos(NumeroComplejo num1, NumeroComplejo num2) {
        double sumaParteReal = num1.parteReal + num2.parteReal;
        double sumaPartelmaginaria = num1.partelmaginaria + num2.partelmaginaria;
        return new NumeroComplejo(sumaParteReal, sumaPartelmaginaria);
    }
    public String toString() {
        if (partelmaginaria >= 0) {
            return parteReal + " + " + partelmaginaria + "i";
        } else {
            return parteReal + " - " + Math.abs(partelmaginaria) + "i";
        }
    }
    public static void main(String[] args) {
        NumeroComplejo num1 = new NumeroComplejo(3, 2);
        NumeroComplejo num2 = new NumeroComplejo(1, -4);
        NumeroComplejo suma = sumaComplejos(num1, num2);
        System.out.println("La suma de los números complejos es: " + suma);
    }
}
```

9.18

Escriba una función en C++ llamada `mayor()` que devuelva la fecha más reciente de cualquier par de fechas que se le transmitan. Por ejemplo, si se transmiten las fechas 10/9/2005 y 11/3/2015 a `mayor()`, será devuelta la segunda fecha.

9.18.1 Código Pseudint

Algoritmo devuelvaFecha

```

Definir fecha, sinespacios, dia, mes, anio Como Cadena
Definir C Como Caracter
Definir i, digitosDia Como Entero
Escribir "FECHA:" Sin Saltar
Leer fecha
C<-Subcadena(fecha, 1, 1)
Si C==" "
    i<-1
    Repetir
        C<-Subcadena(fecha,i,i)
        Si C==" " Entonces
            sinespacios <- Subcadena(fecha,i+1,Longitud(fecha))
        FinSi
        i <- i+1
    Hasta Que !(C==" ")
    fecha <- sinespacios
FinSi
i <- 1
digitosDia <- 0
Repetir
Repetir

```

```

C<-Subcadena(fecha,i,i)
Si !(C=="") Entonces
    digitosDia <- digitosDia + 1

FinSi

i <- i+1
Hasta Que C==" "
sinespacios <- ""

Para i<-1 Hasta Longitud(fecha) Con Paso 1 Hacer

    C<-Subcadena(fecha,i,i)

    Si !(C=="") Entonces
        sinespacios <- sinespacios + Subcadena(fecha,i,i)

    FinSi

FinPara

fecha <- sinespacios
dia <- Subcadena(fecha,1,digitosDia)
mes <- Subcadena(fecha,digitosDia+1,Longitud(fecha)-4)
anio <- Subcadena(fecha,Longitud(fecha)-3,Longitud(fecha))

Escribir dia," " Sin Saltar

Si Mayusculas(mes)=='ENERO' Entonces
    Escribir 1," " Sin Saltar

FinSi

Si Mayusculas(mes)=='FEBRERO' Entonces
    Escribir 2," " Sin Saltar

FinSi

Si Mayusculas(mes)=='MARZO' Entonces
    Escribir 3," " Sin Saltar

FinSi

Si Mayusculas(mes)=='ABRIL' Entonces
    Escribir 4," " Sin Saltar

FinSi

```



```
Si Mayusculas(mes)=='MAYO' Entonces  
  Escribir 5, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='JUNIO' Entonces
```

```
  Escribir 6, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='JULIO' Entonces
```

```
  Escribir 7, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='AGOSTO' Entonces
```

```
  Escribir 8, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='SEPTIEMBRE' Entonces
```

```
  Escribir 9, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='OCTUBRE' Entonces
```

```
  Escribir 10, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='NOVIEMBRE' Entonces
```

```
  Escribir 11, "" Sin Saltar
```

```
FinSi
```

```
Si Mayusculas(mes)=='DICIEMBRE' Entonces
```

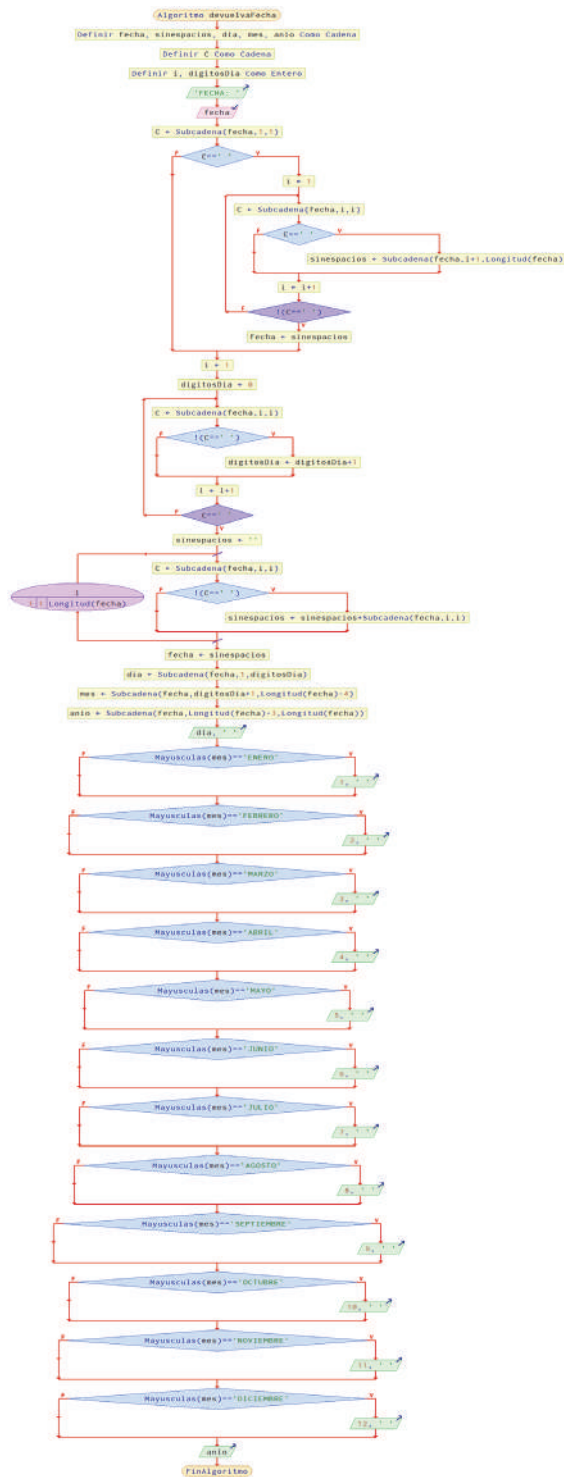
```
  Escribir 12, "" Sin Saltar
```

```
FinSi
```

```
  Escribir anio
```

FinAlgoritmo

9.18.2 Diagrama de flujo



9.18.3 Adaptación JAVA

```
import java.time.LocalDate;
public class FechaMayor {
    public static LocalDate mayor(LocalDate fecha1, LocalDate fecha2) {
        if (fecha1.isAfter(fecha2)) {
            return fecha1;
        } else {
            return fecha2;
        }
    }
    public static void main(String[] args) {
        LocalDate fecha1 = LocalDate.of(2005, 10, 9);
        LocalDate fecha2 = LocalDate.of(2015, 11, 3);
        LocalDate fechaMasReciente = mayor(fecha1, fecha2);
        System.out.println("La fecha más reciente es:" + fechaMasReciente);
    }
}
```

9.19

Realice una función recursiva que sume los primeros n enteros positivos. Nota: para plantear la función recursiva tenga en cuenta que la suma puede expresarse mediante la siguiente recurrencia: $\text{suma}(n) = 1$, si $n=1$ $n+\text{suma}(n-1)$, si $n>1$.

9.19.1 Código Pseint

Funcion SumarEnteros(n)

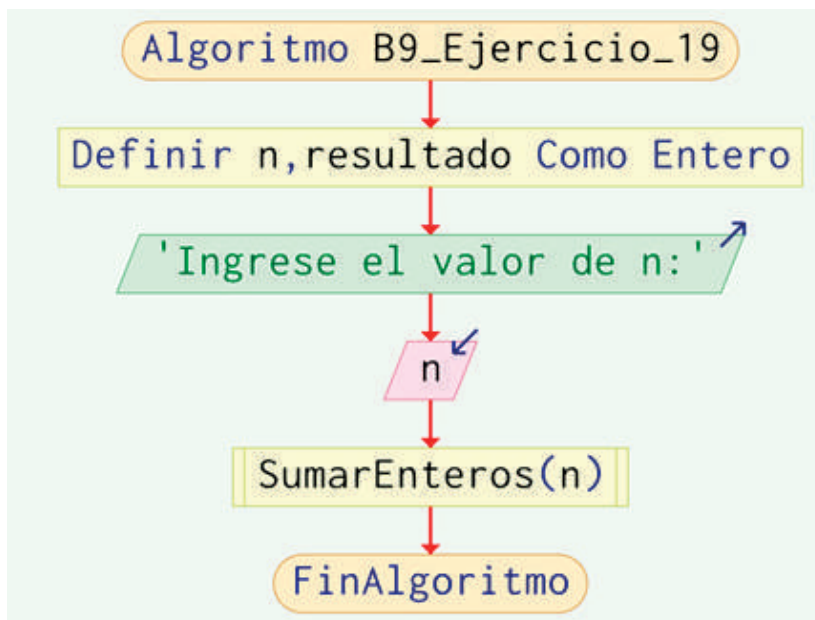
```
suma <- 0
Para i <- 1 Hasta n Hacer
    suma <- suma + i
FinPara
Escribir "La suma de los primeros ", n, " enteros positivos es:", suma
FinFuncion
```

Algoritmo B9_Ejercicio_19

```
Definir n, resultado Como Entero
Escribir "Ingrese el valor de n:"
Leer n
SumarEnteros(n)
```

FinAlgoritmo

9.19.2 Diagrama de flujo



9.19.3 Adaptación JAVA

```
public class Main{  
    public static int sumaEnteros(int n) {  
        if (n == 1) {  
            return 1;  
        } else {  
            return n + sumaEnteros(n - 1);  
        }  
    }  
    public static void main(String[] args) {  
        int n = 5;  
        int resultado = sumaEnteros(n);  
        System.out.println("La suma de los primeros " + n + " enteros positivos es:" +  
            resultado);  
    }  
}
```

9.20

Realice una función recursiva que sume los primeros n enteros positivos. Nota: para plantear la función recursiva tenga en cuenta que la suma puede expresarse mediante la siguiente recurrencia: $\text{suma}(n) = 1$, si $n=1$; $n+\text{suma}(n-1)$, si $n>1$

9.20.1 Código Pseint

Funcion fib <- Fibonacci (TerminoFibo)

Si TerminoFibo==1 o TerminoFibo==2 Entonces

fib=1

SiNo

fib=Fibonacci(TerminoFibo-1)+Fibonacci(TerminoFibo-2)

FinSi

Fin Funcion

Algoritmo Fibonacci_rekursivo

Definir TerminoFibo Como Entero

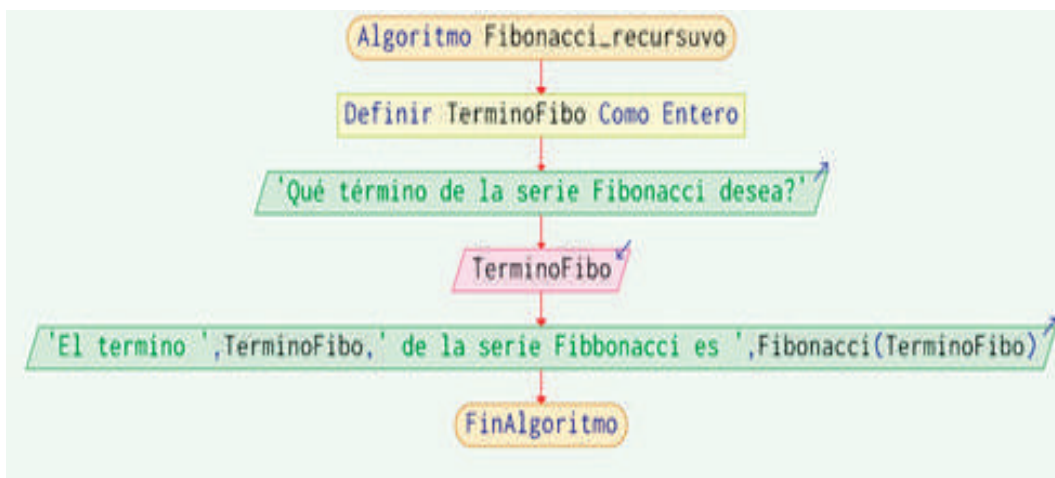
Escribir "Qué término de la serie Fibonacci desea?"

Leer TerminoFibo

Escribir "El termino ',TerminoFibo,' de la serie Fibbonacci es ',Fibonacci(TerminoFibo)

FinAlgoritmo

9.20.2 Diagrama de flujo



9.20.3 Adaptación JAVA

```
public class Main{  
    public static int fibonacci(int n) {  
        if (n == 0) {  
            return 0;  
        } else if (n == 1) {  
            return 1;  
        } else {  
            return fibonacci(n - 1) + fibonacci(n - 2);  
        }  
    }  
    public static void main(String[] args) {  
        int limite = 10;  
        System.out.println("Serie de Fibonacci hasta el término " + limite + ":");  
        for (int i = 0; i <= limite; i++) {  
            System.out.print(fibonacci(i) + " ");  
        }  
    }  
}
```

9.21

Escriba una función recursiva que calcule un número elevado a una potencia entera mayor o igual que cero: x^y .

9.21.1 Código Pseint

Funcion CalcularPotencia(base, exponente)

```
resultado <- 1
```

```
Si exponente >= 0 Entonces
```

```
  Para i <- 1 Hasta exponente Hacer
```

```
    resultado <- resultado * base
```

```
  FinPara
```

```
Sino
```

```
  Para i <- 1 Hasta -exponente Hacer
```

```
    resultado <- resultado / base
```

```
  FinPara
```

```
FinSi
```

```
  Escribir "El resultado de ", base, "^", exponente, " es:", resultado
```

```
FinFuncion
```

Algoritmo B9_Ejercicio_21

```
  Definir base, exponente Como Real
```

```
  Definir resultado Como Real
```

```
  Escribir "Ingrese la base:"
```

```
  Leer base
```

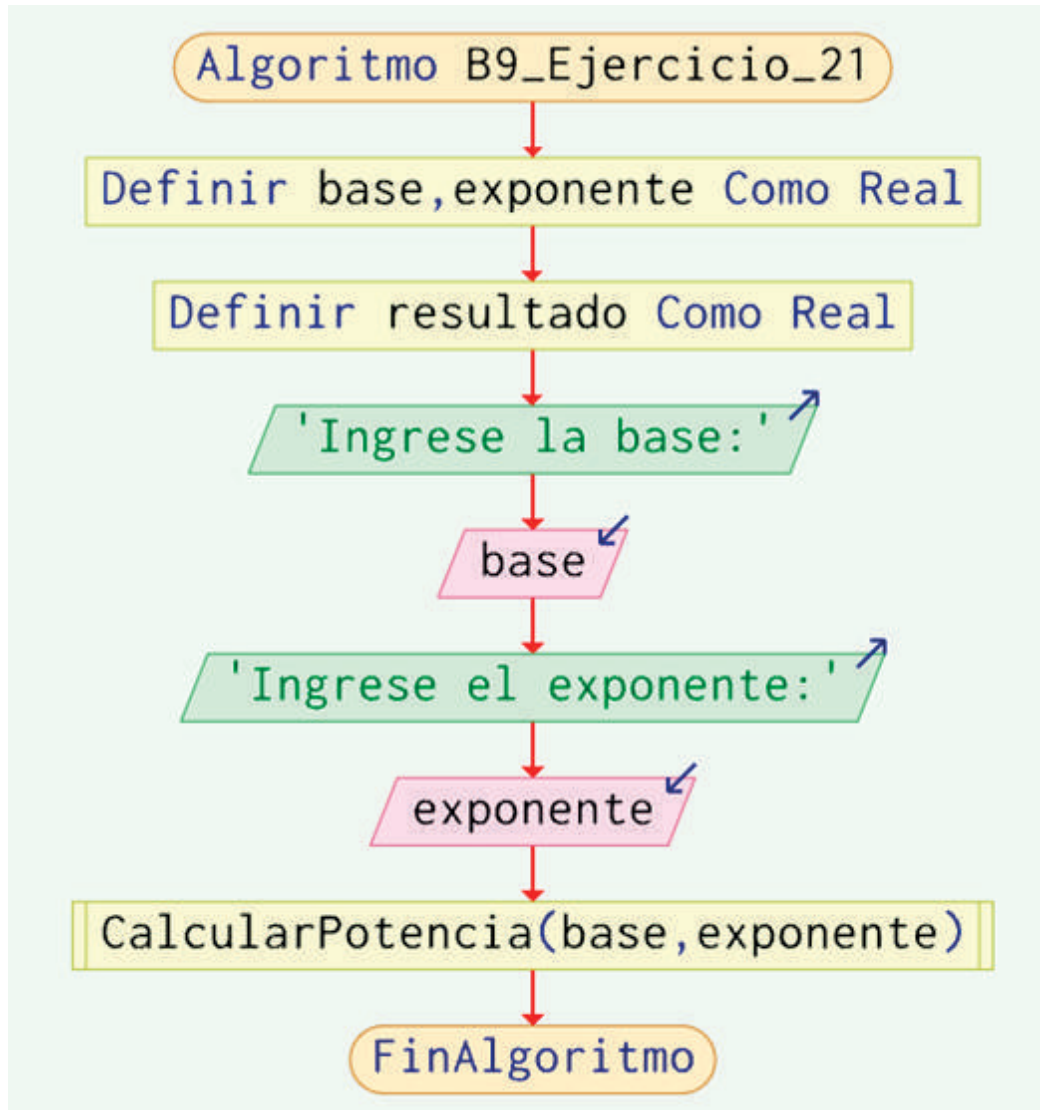
```
  Escribir "Ingrese el exponente:"
```

```
  Leer exponente
```

```
  CalcularPotencia(base, exponente)
```

FinAlgoritmo

9.21.2 Diagrama de flujo



9.21.3 Adaptación JAVA

```
public class Main {  
    public static double potencia(double x, int y) {  
        if (y == 0) {  
            return 1;  
        } else if (y > 0) {  
            return x * potencia(x, y - 1);  
        } else {  
            return (1 / x) * potencia(x, y + 1);  
        }  
    }  
    public static void main(String[] args) {  
        double base = 2.0;  
        int exponente = 3;  
        double resultado = potencia(base, exponente);  
        System.out.println(base + " elevado a la potencia " + exponente + " es:" +  
            resultado);  
    }  
}
```

9.22

Escriba una función `escribeNumeros(int ini, int fin)` que devuelva en la salida estándar los números del inicio al fin. Escriba una versión que escriba los números en orden ascendente

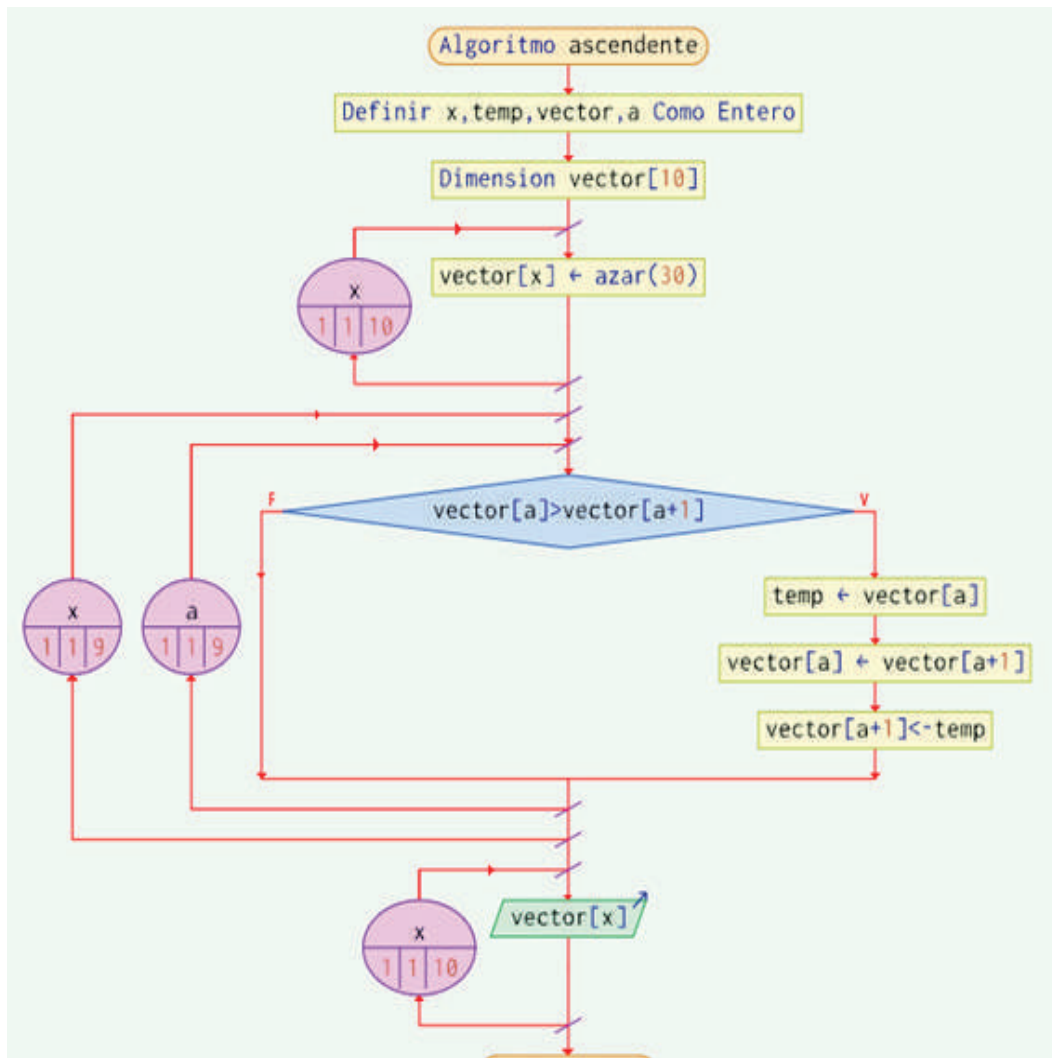
9.22.1 Código Pseint

Algoritmo ascendente

```
Definir x,temp,vector,a Como Entero
Dimension vector[10]
para x = 1 Hasta 10 Con Paso 1 Hacer
    vector[x] = azar(30)
FinPara
para x = 1 Hasta 9 Con Paso 1 Hacer
    para a = 1 Hasta 9 Con Paso 1 Hacer
        si vector[a] > vector[a+1] Entonces
            temp = vector[a]
            vector[a] = vector[a+1]
            vector[a+1] = temp
        FinSi
    FinPara
FinPara
para x = 1 Hasta 10 Con Paso 1 Hacer
    Escribir vector[x]
FinPara
```

FinAlgoritmo

9.22.2 Diagrama de flujo



9.22.3 Adaptación JAVA

```
public class EscribeNumeros {  
    public static void escribeNumerosAscendente(int ini, int fin) {  
        if (ini <= fin) {  
            System.out.print(ini + " ");  
            escribeNumerosAscendente(ini + 1, fin);  
        }  
    }  
  
    public static void escribeNumerosDescendente(int ini, int fin) {  
        if (ini >= fin) {  
            System.out.print(ini + " ");  
            escribeNumerosDescendente(ini - 1, fin);  
        }  
    }  
  
    public static void main(String[] args) {  
        int inicio = 1;  
        int fin = 10;  
  
        System.out.println("Números en orden ascendente:");  
        escribeNumerosAscendente(inicio, fin);  
  
        System.out.println("\nNúmeros en orden descendente:");  
        escribeNumerosDescendente(fin, inicio);  
    }  
}
```

Bloque 10: Pilas

10.1.

Hacer un programa para agregar números enteros a una pila, hasta que el usuario lo decida, después mostrar todos los números introducidos en la pila.

10.1.1. Código Pseint

Algoritmo B I I_Ejercicio_I

Definir numeros Como Entero

Definir opcion Como Caracter

Definir listaNumeros Como Cadena

listaNumeros <- ""

Repetir

 Escribir "Ingrese un número:"

 Leer numeros

 listaNumeros <- listaNumeros + "" + ConvertirATexto(numeros)

 Escribir "¿Desea ingresar otro número? (s/n):"

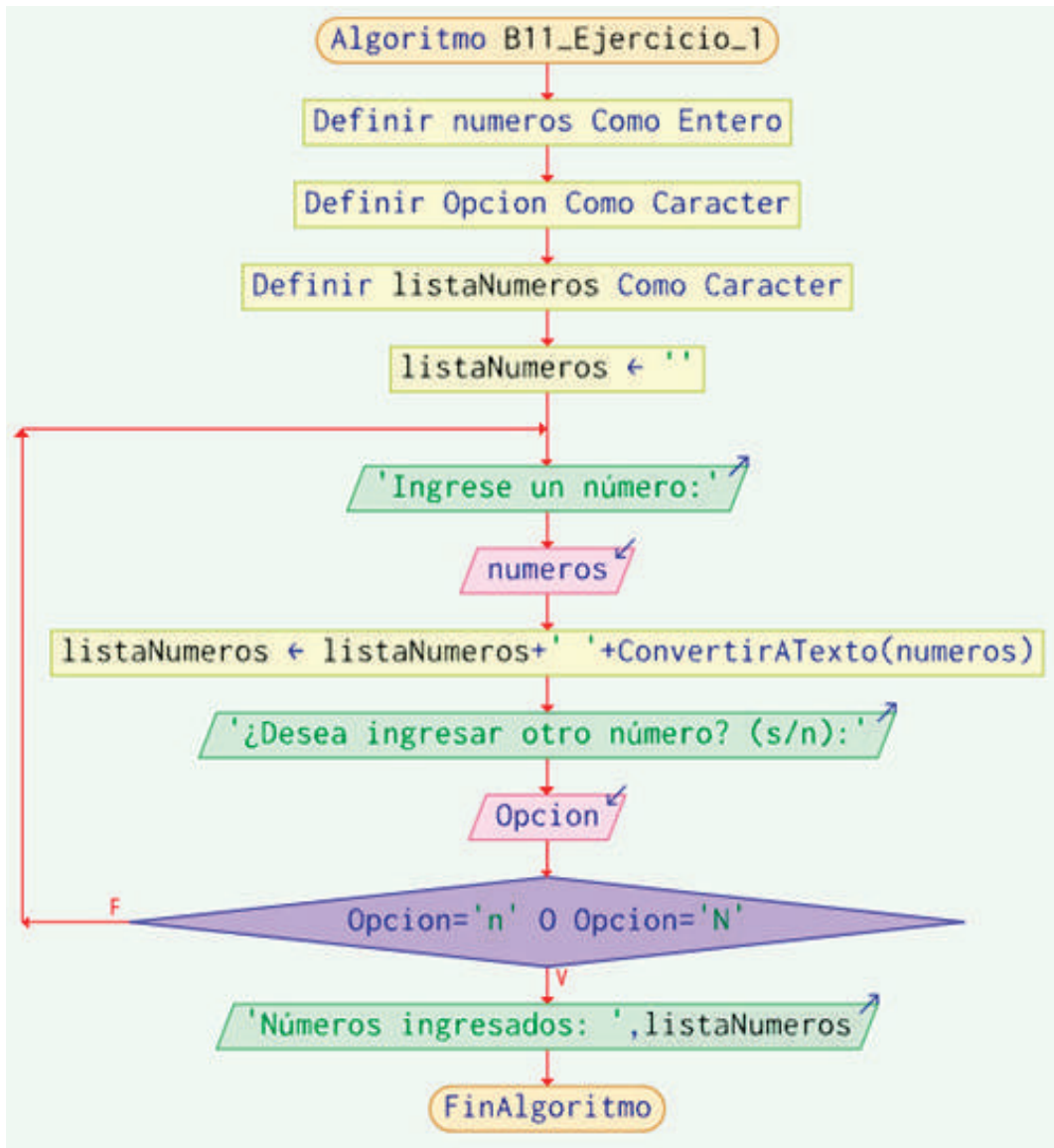
 Leer opcion

Hasta Que opcion = "n" o Opcion = "N"

Escribir "Números ingresados:", listaNumeros

FinAlgoritmo

10.1.2. Diagrama de flujo



10.1.3. Adaptación a Java

```
import java.util.Stack;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        Stack<Integer> pila = new Stack<Integer>();
        int valor, opcion = 1, contador;
        while(opcion == 1){
            valor = Integer.parseInt(JOptionPane.showInputDialog(null,"Digite el valor a
            ingresar", "Digite un valor entero"));
        pila.push(valor);
            contador = 0;
        do{
            if(contador > 0){
                JOptionPane.showMessageDialog(null, "ERROR, Emplee los valores en
                el rango");
            }
            opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Desea
            continuar? (1) SI (2) NO"));
            contador++;
        } while (opcion < 1 || opcion > 2);
        }
        JOptionPane.showMessageDialog(null, "Los números introducidos en la pila son:
        ");
        System.out.println("Pila: " + pila);
        JOptionPane.showMessageDialog(null, "Cerrando...");
    }
}
```


10.2.2

Hacer un programa en Java, utilizando Pilas que contenga el siguiente menú:

1. Insertar un carácter a la pila
2. Mostrar todos los elementos de la pila
3. Salir

10.2.1. Código Pseint

Algoritmo Pilas

```
Definir pila como entero;
Definir max como entero;
Definir tope como entero;
Definir seleccion como entero;
Definir dato como entero;
Definir band como logico;

tope = 0;
band = Falso;
max = 11;
dato = 0;
Dimension pila[max];

Repetir
    Escribir "Seleccione que quiere hacer con la estructura pila (Tope = ",tope,")";
    Escribir "1 = Saber si la pila está vacía";
    Escribir "2 = Saber si la pila está llena";
    Escribir "3 = Colocar un elemento a la pila";
    Escribir "4 = Quitar un elemento a la pila";
    Escribir "5 = Mostrar los elementos actuales de la pila";
    Escribir "Cualquier otro número = Salir";
```

Leer seleccion;

Segun seleccion Hacer

1:

band = pilaVacia(pila,tope);

Si (band=Verdadero) Entonces

Escribir “La pila está vacía”;

Sino

Escribir “La pila no está vacía”;

FinSi

2:

band = pilaLlena(pila,tope,max);

Si (band=Verdadero) Entonces

Escribir “La pila está llena”;

Sino

Escribir “La pila no está llena”;

FinSi

3:

tope = poner(pila,tope,max,dato);

4:

tope = quitar(pila, tope,max);

5:

mostrarElementos(pila);

De Otro Modo:

Escribir “Hasta luego”;

Fin Segun

Escribir “ ”;

Hasta Que(seleccion <> 1 y seleccion <> 2 y seleccion <> 3 y seleccion <> 4
y seleccion <> 5)

FinAlgoritmo

```
Funcion band <- pilaVacía(pila,tope)
    Si(tope=0) Entonces
        band <- Verdadero
    Sino
        band <- Falso;
    FinSi
FinFuncion

Funcion band <- pilaLlena(pila,tope,max)
    Si(tope=max) Entonces
        band <- Verdadero;
    Sino
        band <- falso;
    FinSi
FinFuncion

Funcion tope <- poner(pila,tope,max,dato)
    band = pilaLlena(pila,tope,max);
    Si(band=Verdadero) Entonces
        Escribir “Desbordamiento - Pila llena”;
    Sino
        Escribir Sin Saltar “Coloque dato NUMÉRICO a insertar”; Leer dato;
        pila[tope] = dato;
        tope <- tope + 1;
    FinSi
FinFuncion

Funcion tope <- quitar(pila, tope,max)
    band = pilaVacía(pila,tope);
```

Si(band=Verdadero) Entonces

 Escribir “Subdesbordamiento - Pila vacía”;

Sino

 tope <- tope-1;

 pila[tope] = 0;

FinSi

FinFuncion

Funcion mostrarElementos(pila)

 Escribir “La pila tiene los siguientes elementos actualmente:”;

 Escribir “Posición 11:”, pila[10];

 Escribir “Posición 10:”, pila[9];

 Escribir “Posición 9:”, pila[8];

 Escribir “Posición 8:”, pila[7];

 Escribir “Posición 7:”, pila[6];

 Escribir “Posición 6:”, pila[5];

 Escribir “Posición 5:”, pila[4];

 Escribir “Posición 4:”, pila[3];

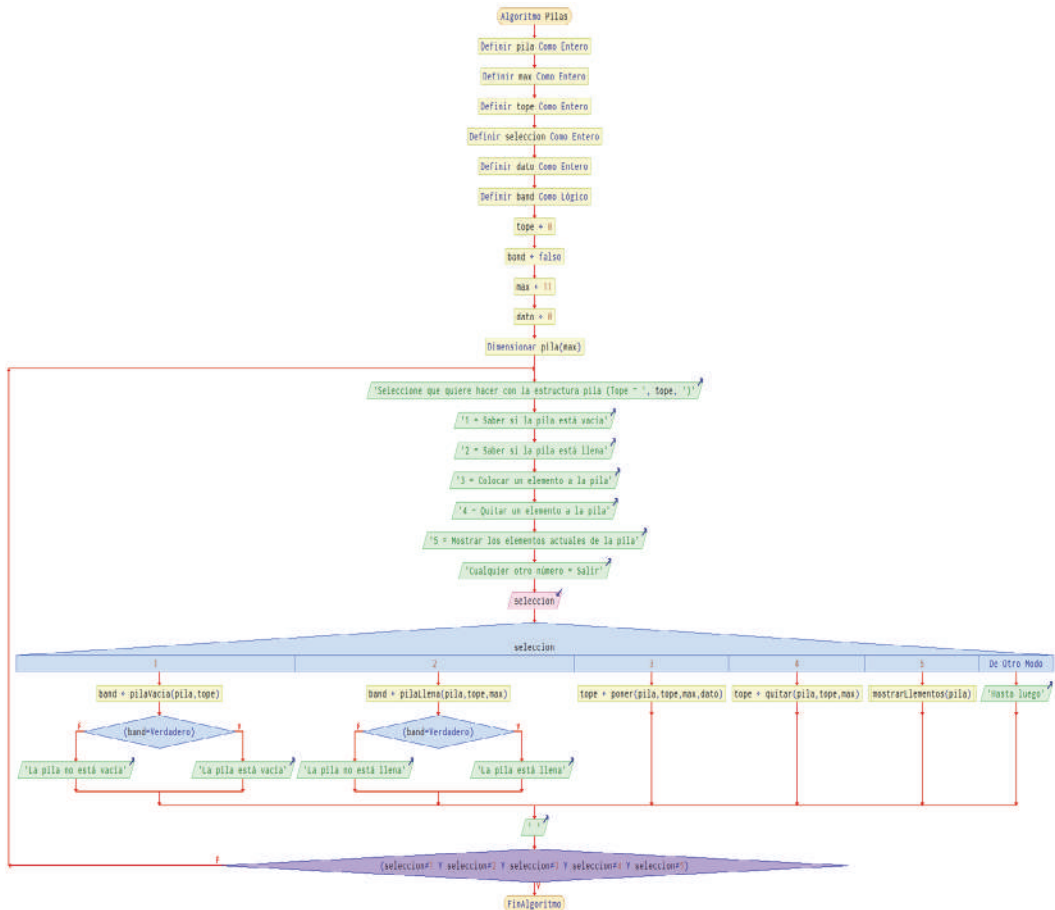
 Escribir “Posición 3:”, pila[2];

 Escribir “Posición 2:”, pila[1];

 Escribir “Posición 1:”, pila[0];

FinFuncion

10.3.2. Diagrama de flujo



10.2.3 Adaptación a Java

```

import javax.swing.JOptionPane;
import java.util.Stack;

public class Ejercicio2Bloque10 {

    public static void main(String[] args) {

        Stack<Character> pila = new Stack<>();

        int opcion;
        do {
            String menu = "Menú:\n"

```

```
+ "1. Insertar un carácter a la pila\n"  
+ "2. Mostrar todos los elementos de la pila\n"  
+ "3. Salir\n\n"  
+ "Ingrese su opción:"  
  
opcion = Integer.parseInt(JOptionPane.showInputDialog(null, menu));  
switch (opcion) {  
  
    case 1 -> {  
        String caracterString = JOptionPane.showInputDialog("Ingrese un  
carácter:");  
  
        char caracter = caracterString.charAt(0);  
  
        pila.push(caracter);  
  
        JOptionPane.showMessageDialog(null, "Carácter agregado a la pila.");  
    }  
    case 2 -> {  
        StringBuilder elementos = new StringBuilder("Elementos en la pila:\n");  
  
        for (char elemento : pila) {  
            elementos.append(elemento).append("\n");  
        }  
  
        JOptionPane.showMessageDialog(null, elementos  
            elementos.append(elemento).append("\n");  
        }  
  
        JOptionPane.showMessageDialog(null, elementos.toString());  
    }  
  
    case 3 -> JOptionPane.showMessageDialog(null, "¡Hasta luego!");  
    default -> JOptionPane.showMessageDialog(null, "Opción inválida. Intente  
nuevamente.");  
    }  
} while (opcion != 3);  
}
```

10.3

Hacer un ejercicio en Java que agregue números enteros a una pila, los muestre, pero también mencione cuál es el mayor de ellos.

10.3.2 Pseudo código: Reto

Para el presente ejercicio, el reto propuesto es desarrollar el pseudo código y diagrama de flujo en Pseint

10.3.3 Adaptación a Java

```
import java.util.Stack;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        Stack<Integer> pila = new Stack<Integer>();
        int valor, opcion = 1, contador, mayor = Integer.MIN_VALUE, mayorAux;
        while(opcion == 1){
            valor = Integer.parseInt(JOptionPane.showInputDialog(null,"Digite el valor a
            ingresar", "Digite un valor entero"));
            pila.push(valor);
            // Continuar nuevamente
            contador = 0;
            do{
                if(contador > 0){
                    JOptionPane.showMessageDialog(null,"ERROR, Emplee los valores en
                    el rango");
                }
                opcion = Integer
```

```
opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Desea continuar?  
(1) SI (2) NO"));  
  
    contador++;  
  
    if(pila.peek() > mayor){  
        mayor = pila.peek();  
    }  
  
} while (opcion < 1 || opcion > 2);  
  
}  
  
JOptionPane.showMessageDialog(null, "Los números introducidos en la pila son:  
");  
  
System.out.println("Pila: " + pila);  
  
System.out.println("Valor mas alto: " + mayor);  
  
JOptionPane.showMessageDialog(null, "Cerrando...");  
  
}  
  
}
```

104.

Hacer un programa en Java que agregue números enteros a una pila, los muestre y calcule la suma de todos ellos:

104.2 Pseint: Reto propuesto

Para el presente ejercicio, se sugiere, a manera de reto, desarrollar el ejercicio con Pseint.

104.3 Adaptación a Java

```
import javax.swing.JOptionPane;
import java.util.Stack;
public class Ejercicio4Bloque10 {
    public static void main(String[] args) {
        Stack<Double> pila = new Stack<>();
        double suma = 0.0;
        int cantidadNumeros = Integer.parseInt(JOptionPane.showInputDialog("Ingrese
la cantidad de números a agregar:"));
        for (int i = 0; i < cantidadNumeros; i++) {
            double numero = Double.parseDouble(JOptionPane.showInputDialog("In-
grese el número #" + (i + 1) + ":"));
            pila.push(numero);
        }
        StringBuilder elementosPila = new StringBuilder("Elementos en la pila:\n");
        for (double elemento : pila) {
            elementosPila.append(elemento).append("\n");
            suma += elemento;
        }
        JOptionPane.showMessageDialog(null, elementosPila.toString());
        JOptionPane.showMessageDialog(null, "Suma de los elementos:" + suma);
    }
}
```

Bloque 11: Colas

11.2.

Hacer un programa en JAVA, utilizando Colas que contenga el siguiente menú:

1. Insertar un carácter a una cola
2. Mostrar todos los elementos de la cola
3. Salir

11.2.1 Código Pseint

Algoritmo B12_Ejercicio_1

Definir opcion Como Entero

Definir caracteres Como Cadena

caracteres <- ""

Repetir

 Escribir "Menú:"

 Escribir "1. Insertar un carácter"

 Escribir "2. Mostrar todos los elementos ingresados"

 Escribir "3. Salir"

 Escribir "Ingrese la opción:"

 Leer opcion

Segun opcion Hacer

1:

Escribir "Ingrese un carácter:"
 Leer caracter
 $\text{caracteres} \leftarrow \text{caracteres} + \text{caracter}$

2:

Escribir "Elementos ingresados:"
 Escribir caracteres

3:

Escribir "Saliendo del programa..."

De Otro Modo:

Escribir "Opción inválida. Por favor, ingrese una opción válida."

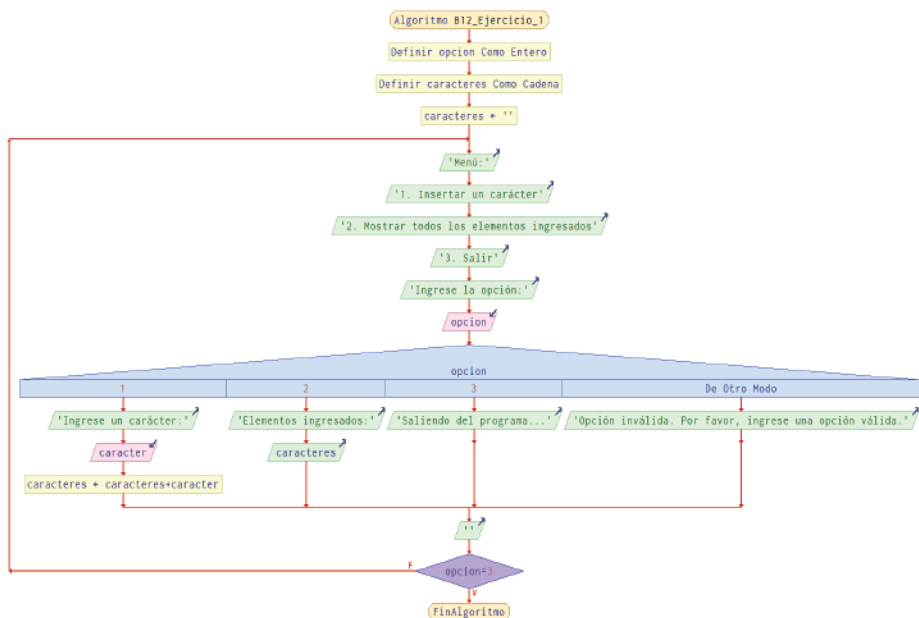
FinSegun

Escribir ""

Hasta Que opcion = 3

FinAlgoritmo

11.2.2 Diagrama de flujo



11.2.3. Adaptación a Java

```
import java.util.LinkedList;

import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args){

        char character;

        int opcion, contadorError, continuar = 1;

        LinkedList cola = new LinkedList();

        JOptionPane.showMessageDialog(null, "Bienvenido al Menú :D");

        do{
            do{
                opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Qué operación desea realizar?"));

                +"\n1. Insertar un carácter a la cola"

                +"\n2. Mostrar todos los elementos de la cola"

                +"\n3. Salir", "Ingrese dato...");

                if(opcion < 1 || opcion > 3){
                    JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
                }
            }while(opcion < 1 || opcion > 3);

            switch(opcion){

                case 1:
                    String entrada = JOptionPane.showInputDialog("Digite el caracter:");
                    character = entrada.charAt(0);

                    cola.offer(character);

                    break;

                case 2:
                    System.out.println("Todos los elementos de la cola son:");
```

```
        while(cola.peek() != null){
            System.out.println(cola.poll());
        }

        break;

    case 3:
        JOptionPane.showMessageDialog(null, "Finalizado :D");
        break;
    }
}while(opcion != 3);
}
```

11.2

Hacer un programa que guarde datos de clientes de un banco, los almacene en cola, y por último muestre los clientes en el orden correcto.

11.2.1 Código Pseint

Algoritmo banco

```
total <- 0;

Escribir Sin Saltar "Ingresa el valor de n:";

leer n;

Para i<-1 Hasta n Con Paso 1 Hacer

    Escribir "PROCESO ", i;

    Escribir Sin Saltar "Ingresa el valor de cantidad 01 de enero:";

    Leer cantidad_01_enero;
    Escribir Sin Saltar "Ingresa el valor de cantidad 02 de febrero:";

    Leer cantidad_02_febrero;
    Escribir Sin Saltar "Ingresa el valor de cantidad 03 de marzo:";

    Leer cantidad_03_marzo;
    Escribir Sin Saltar "Ingresa el valor de cantidad 04 de abril:";
```

```
Leer cantidad_04_abril;
Escribir Sin Saltar "Ingresa el valor de cantidad 05 de mayo:";

Leer cantidad_05_mayo;
Escribir Sin Saltar "Ingresa el valor de cantidad 06 de junio:";

Leer cantidad_06_junio;
Escribir Sin Saltar "Ingresa el valor de cantidad 07 de julio:";

Leer cantidad_07_julio;
Escribir Sin Saltar "Ingresa el valor de cantidad 08 de agosto:";

Leer cantidad_08_agosto;
Escribir Sin Saltar "Ingresa el valor de cantidad 09 de septiembre:";

Leer cantidad_09_septiembre;
Escribir Sin Saltar "Ingresa el valor de cantidad 10 de octubre:";

Leer cantidad_10_octubre;
Escribir Sin Saltar "Ingresa el valor de cantidad 11 de noviembre:";

Leer cantidad_11_noviembre;
Escribir Sin Saltar "Ingresa el valor de cantidad 12 de diciembre:";

Escribir Sin Saltar "Ingresa el valor de cantidad 12 de diciembre:";
Leer cantidad_12_diciembre;

interes <- total*0.1

total <- total+interes+cantidad_01_enero+cantidad_02_febrero+cantidad_03_
marzo+cantidad_04_abril+cantidad_05_mayo+cantidad_06_junio+canti-
dad_07_julio+cantidad_08_agosto+cantidad_09_septiembre+cantidad_10_oc-
tubre+cantidad_11_noviembre+cantidad_12_diciembre;

inversion_final <- total;

Escribir "Valor de intereses:", interes;

Escribir "Valor de inversion final:", inversion_final;

Escribir "";

FinPara

Escribir "Valor de total:", total;
```

FinAlgoritmo

11.2.2. Diagrama de flujo



11.2.3. Adaptación a Java

```
import java.util.LinkedList;
import java.util.Queue;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args) {
        Queue<String> colaClientes = new LinkedList<>();
        int opcion = 0;
        while (opcion != 3) {
            String menu = "Menú:\n"
                + "1. Agregar cliente\n"
                + "2. Mostrar clientes\n"
                + "3. Salir\n\n"
                + "Ingrese la opción deseada:";
            opcion = Integer.parseInt(JOptionPane.showInputDialog(menu));
            switch (opcion) {
                case 1:
                    String nombreCliente = JOptionPane.showInputDialog("Ingrese el nombre del cliente:");
                    colaClientes.offer(nombreCliente);
                    JOptionPane.showMessageDialog(null, "Cliente agregado correctamente.");
                    break;
                case 2:
                    StringBuilder clientes = new StringBuilder("Clientes:\n");
                    for (String cliente : colaClientes) {
                        clientes.append(cliente).append("\n");
                    }
            }
        }
    }
}
```



```
        JOptionPane.showMessageDialog(null, clientes.toString());
        break;
    case 3:
        JOptionPane.showMessageDialog(null, "Saliendo del programa...");
        break;
    default:
        JOptionPane.showMessageDialog(null, "Opción inválida. Intente nuevamente.");
        break;
    }
}
}
```

11.3

Realizar un programa en Java, aplicando Colas, que inserte números enteros a una cola, y que, al salir, muestre el mayor y menor.

11.3.1 Pseint: Ejercicio propuesto

Para este ejercicio, se sugiere intentarlo a manera de práctica en Pseint.

11.3.2 Adaptación a Java

```
import java.util.LinkedList;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        char caracter;

        int opcion, contadorError, continuar = 1, mayor = Integer.MIN_VALUE, menor
        = Integer.MAX_VALUE;
```

```
LinkedList <Integer> cola = new LinkedList();
JOptionPane.showMessageDialog(null, "Bienvenido al Menú :D");
do{
    do{
        opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Qué operación desea realizar?");
        +"\n1. Insertar un número entero a la cola"
        +"\n2. Salir", "Ingrese dato...");
        if(opcion < 1 || opcion > 2){
            JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
        }
    }while(opcion < 1 || opcion > 2);
    switch(opcion){
        case 1:
            int entrada = Integer.parseInt(JOptionPane.showInputDialog("Digite el número:"));
            cola.offer(entrada);
            break;
        case 2:
            JOptionPane.showMessageDialog(null, "Finalizando...");
            break;
    }
}while(opcion != 2);
for (Integer valor : cola){
    if(valor > mayor){
        mayor = valor;
    }
}
```

```
for (Integer valor : cola){
    if(valor < menor){
        menor = valor;
    }
}
System.out.println("El elemento mayor es:" + mayor);
System.out.println("El elemento menor es:" + menor);
}
```

114.

Realizar un programa en Java, aplicando colas, que deje ingresar solo números primos y que use el siguiente menú:

- 1) Ingresar número Primo.
- 2) Mostrar los números primos y la suma de todos ellos.
- 3) Salir del Programa.

114.1 Pseint: Ejercicio propuesto

Para el ejercicio, se propone a manera de reto desarrollarlo en Pseint.

114.2 Adaptación a Java

```
import java.util.LinkedList;
import java.util.Queue;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args) {
```

```
Queue<Integer> colaNumerosPrimos = new LinkedList<>();

int opcion = 0;
int suma = 0;
while (opcion != 3) {

    String menu = "Menú:\n"
        + "1. Insertar número\n"
        + "2. Mostrar los números primos y la suma de todos ellos.\n"
        + "3. Salir\n\n"
        + "Ingrese la opción deseada:";

    opcion = Integer.parseInt(JOptionPane.showInputDialog(menu));

    switch (opcion) {

        case 1:
            int numero = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un número entero:"));

            if (esPrimo(numero)) {

                colaNumerosPrimos.offer(numero);

                suma += numero;

                JOptionPane.showMessageDialog(null, "Número primo insertado correctamente.");
            } else {

                JOptionPane.showMessageDialog(null, "El número no es primo. Intente con otro número.");
            }

            break;

        case 2:
            StringBuilder output = new StringBuilder("Números primos en la cola:\n");

            Queue<Integer> copiaCola = new LinkedList<>(colaNumerosPrimos);

            while (!copiaCola.isEmpty()) {

                output.append(copiaCola.poll()).append("\n");
            }
    }
}
```

```
output.append("\nSuma de los números:").append(suma);
    JOptionPane.showMessageDialog(null, output.toString());
    break;
case 3:
    JOptionPane.showMessageDialog(null, "Saliendo del programa...");
    break;
default:
    JOptionPane.showMessageDialog(null, "Opción inválida. Intente nuevamente.");
    break;
}
}
}
public static boolean esPrimo(int numero) {
    if (numero <= 1) {
        return false;
    }
    for (int i = 2; i <= Math.sqrt(numero); i++) {
        if (numero % i == 0) {
            return false;
        }
    }
    return true;
}
}
```

Bloque 12: Listas

12.1.

Crear una lista que almacene “n” números enteros y calcular el menor y mayor de ellos.

12.1.1 Código Pseint

Algoritmo B13_Ejercicio_1

Definir n Como Entero

Definir i, menor, mayor Como Entero

Escribir “Ingrese la cantidad de números a almacenar:”

Leer n

Dimension numeros[n]

Escribir “Ingrese los números:”

Para i <- 1 Hasta n Hacer

 Escribir “Número “, i, “:”

 Leer numeros[i]

FinPara

menor <- numeros[1]

mayor <- numeros[1]

Para $i \leftarrow 2$ Hasta n Hacer

Si $\text{numeros}[i] < \text{menor}$ Entonces

$\text{menor} \leftarrow \text{numeros}[i]$

FinSi

Si $\text{numeros}[i] > \text{mayor}$ Entonces

$\text{mayor} \leftarrow \text{numeros}[i]$

FinSi

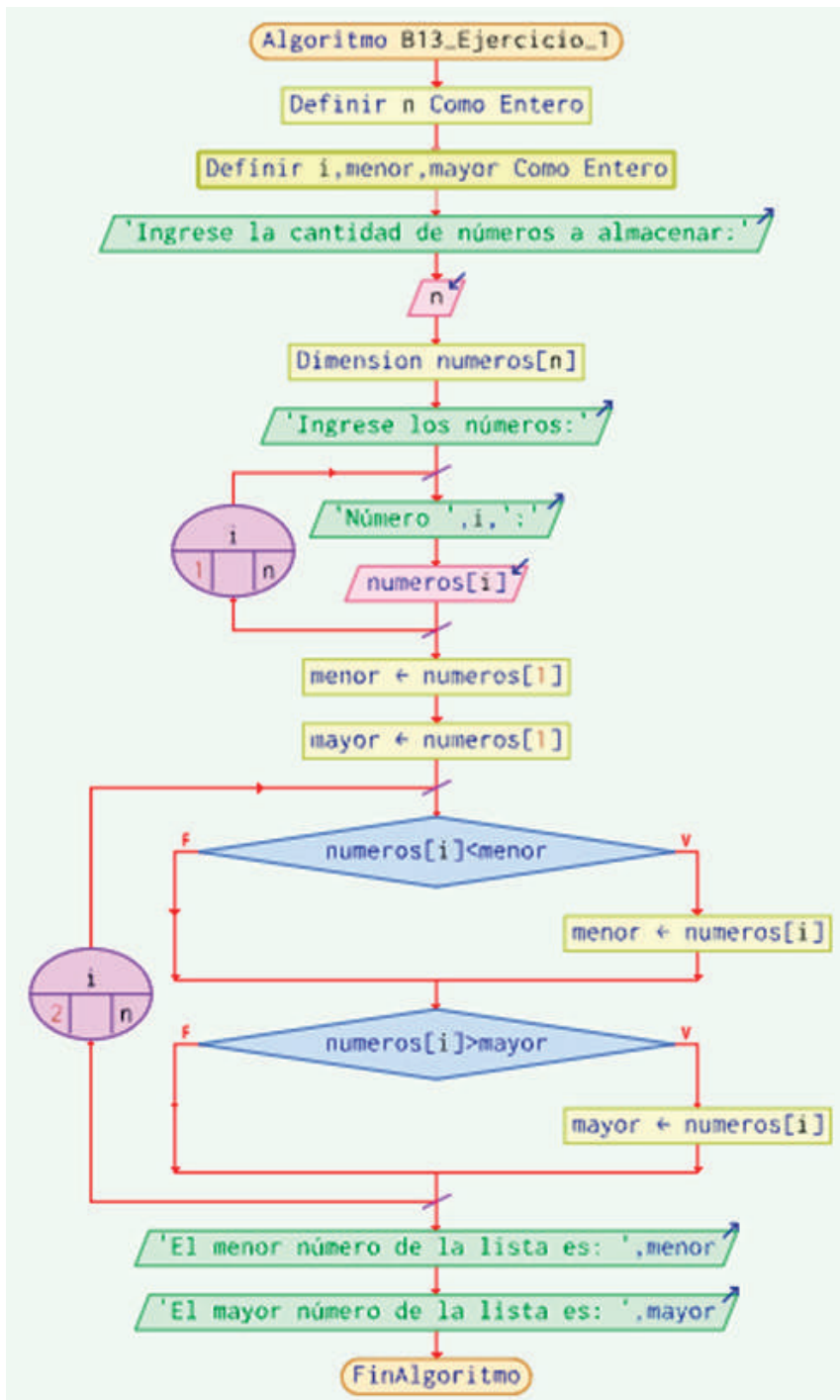
FinPara

Escribir "El menor número de la lista es:", menor

Escribir "El mayor número de la lista es:", mayor

FinAlgoritmo

12.1.2 Diagrama de flujo



12.1.3 Adaptación a Java

```
package Ejercicio I;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args){
        Lista listita = new Lista();
        int opcion = 0, el, inicio = 0;
        do{
            try{
                inicio += 1;
                el = Integer.parseInt(JOptionPane.showInputDialog(null, "Elemento:",
                "Ingresando elemento...", 3));
                if(inicio == 1){
                    listita.agregarAlFinal(el);
                } else{
                    listita.agregarAlFinal(el);
                }
                opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Desea
                continuar?: (1) Sí (2) No", "Ingresando el elemento", 3));
            } catch (Exception e){
                JOptionPane.showMessageDialog(null, "Error " + e.getMessage());
            }
        } while(opcion != 2);
        listita.mostrarLista();
    }
}
```

package Ejercicio I;

```
public class Lista {  
    protected Nodo inicio, fin; //Punteros para saber donde está el inicio y fin  
    public Lista(){  
        inicio = null;  
        fin = null;  
    }  
    // Método para saber si la lista está vacía  
    public boolean estaVacia(){  
        if(inicio == null){  
            return true;  
        } else{  
            return false;  
        }  
    }  
    // Método para insertar al final de la lista  
    public void agregarAlFinal(int elemento){  
        if(!estaVacia()){  
            fin.siguiete = new Nodo(elemento);  
            fin = fin.siguiete;  
        }else{  
            inicio = fin = new Nodo(elemento);  
        }  
    }  
    //Método para agregar un Nodo al inicio de la Lista  
    public void agregarAlInicio(int elemento){
```

```
// Creando al nodo
inicio = new Nodo(elemento, inicio);
if (fin == null){
    fin = inicio;
}
}
// Método para mostrar los datos
public void mostrarLista(){
    int menor = 0, posicion = 0, mayor = 0;
    Nodo recorrer = inicio;
    System.out.println("Arreglo:");
    while (recorrer != null){
        System.out.print("[ " + recorrer.dato + " ]--->");
        if(posicion == 0){
            menor = recorrer.dato;
        } else{
            if(menor > recorrer.dato){
                menor = recorrer.dato;
            }
        }
        if(posicion == 0){
            mayor = recorrer.dato;
        } else{
            if(mayor < recorrer.dato){
                mayor = recorrer.dato;
            }
        }
    }
}
```

```
        recorrer = recorrer.siguiete;
        posicion += 1;
    }
    System.out.println();
    System.out.println("El elemento menor es:" + menor);
    System.out.println("El elemento mayor es:" + mayor);
}
}
```

package Ejercicio I;

```
public class Nodo {
    public int dato;
    public Nodo siguiente; //Puntero enlace
    // Constructor para insertar al final
    public Nodo(int dato){
        this.dato = dato;
        this.siguiente = null;
    }
    //constructor para insertar al inicio
    public Nodo(int dato, Nodo n){
        this.dato = dato;
        siguiente = n;
    }
}
```

12.2

Crear una lista que almacene “n” números reales y calcular su suma y promedio.

12.2.1 Pseudo código

Algoritmo NumerosReales

Escribir “Ingrese la cantidad de numeros reales”

leer n

Dimension datos[n,3]

Definir i, j Como Enteros

Definir proAl,suma,proGene Como Real

i<-1

Mientras (i<=n) Hacer

 Escribir “** reales:“,i

 Escribir “Número real 1:“

 leer n1

 Escribir “Número real 2:“

 leer n2

 Escribir “Número real 3:“

 leer n3

 datos[i, 1] <- n1

 datos[i, 2] <- n2

 datos[i, 3] <- n3

 i<-i+1

FinMientras

j<-1

Mientras (j<=n) Hacer

 proAl <- (datos[j, 1] + datos[j,2] + datos[j,3])/3

 Escribir “El promedio de los números reales:“,j,”Es:“,proAl

 suma<-suma+proAl

 j<-j+1

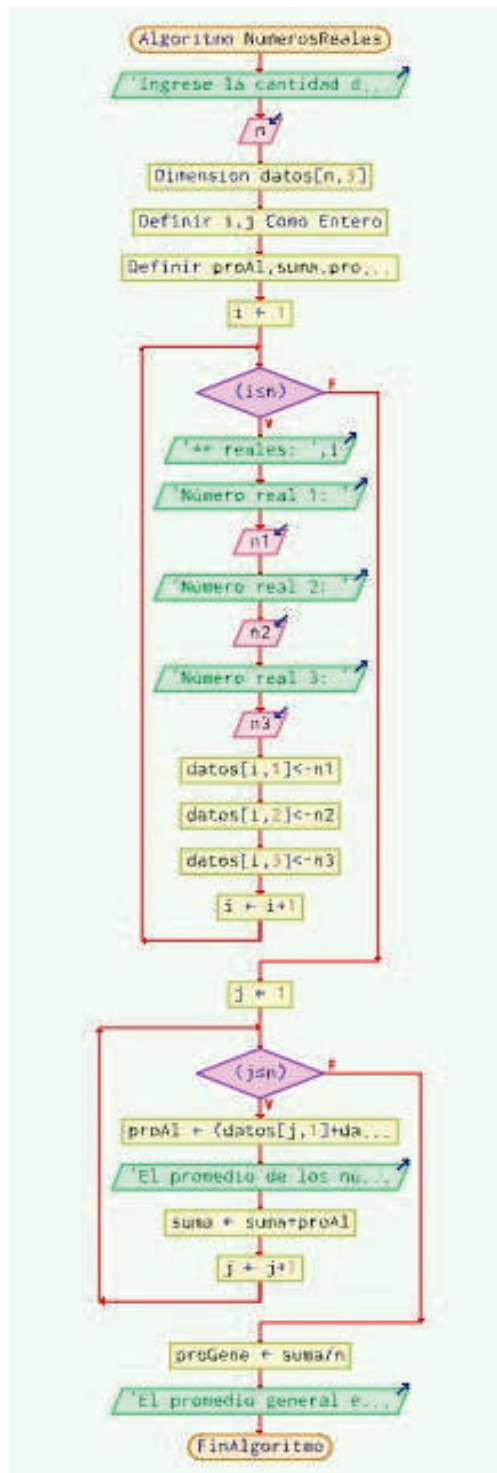
FinMientras

proGene<-suma/n

Escribir “El promedio general es:“,proGene

FinAlgoritmo

12.2.2 Diagrama de flujo



12.2.3 Adaptación a Java

```
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args) {
        List<Double> numeros = new ArrayList<>();

        String cantidadNumerosStr = JOptionPane.showInputDialog("Ingrese la cantidad de números a almacenar:");

        int cantidadNumeros = Integer.parseInt(cantidadNumerosStr);

        for (int i = 0; i < cantidadNumeros; i++) {

            String numeroStr = JOptionPane.showInputDialog("Ingrese el número #" + (i + 1) + " :");

            double numero = Double.parseDouble(numeroStr);

            numeros.add(numero);
        }

        if (numeros.isEmpty()) {

            JOptionPane.showMessageDialog(null, "La lista está vacía. No se puede calcular la suma y el promedio.");

        } else {

            double suma = 0;

            for (double numero : numeros) {

                suma += numero;

            }

            double promedio = suma / numeros.size();

            JOptionPane.showMessageDialog(null, "Suma:" + suma + "\nPromedio:" + promedio);

        }

    }
}
```

12.3

Hacer un programa en JAVA que agregue un elemento al inicio y al final de la lista, muestre los datos de la lista, elimine un elemento del inicio y del final de la lista, y que tenga una sexta opción final para salir.

12.3.1 Pseint: Ejercicio propuesto

Se propone al lector desarrollar el ejercicio en Pseint. Pero cabe destacar que se ofrece una pista de su solución en el lenguaje JAVA.

12.3.2 Adaptación a Java

```
package Simples;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```

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

```
        Lista listita = new Lista();
```

```
        int opcion = 0, el;
```

```
        do{
```

```
            try{
```

```
                opcion = Integer.parseInt(JOptionPane.showInputDialog(null,
```

```
                    "1.Agregar un elemento al inicio de la Lista\n2.Agregar un elemento al final de la Lista\n"
```

```
                    + "3. Mostrar los Datos de la lista\n"
```

```
                    + "4. Eliminar un elemento del inicio de la lista\n"
```

```
                    + "5. Eliminar un elemento del final de la lista\n"
```

```
                    + "6. Salir", "Menú de Opciones", 3));
```

```
                switch(opcion){
```

```
                    case 1:
```

```
                        try{
```



```
el = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingresa el elemento:",
"Insertando al Inicio", 3));

    //Agregando al Nodo
    listita.agregarAlInicio(el);
} catch (NumberFormatException n) {
    JOptionPane.showMessageDialog(null, "Error " + n.getMessage());
}
break;
case 2:
    try {
        el = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingresa
el elemento:", "Insertando al Final", 3));

        //Agregando al Nodo
        listita.agregarAlFinal(el);
    } catch (NumberFormatException n) {
        JOptionPane.showMessageDialog(null, "Error " + n.getMessage());
    }
    break;
case 3:
    listita.mostrarLista();
    break;
case 4:
    el = listita.borrarDelInicio();
    JOptionPane.showMessageDialog(null, "El elemento eliminado es:" + el,
"Eliminando Nodos del inicio", JOptionPane.INFORMATION_MESSAGE);
    break;
```

```
        case 5:
            el = listita.borrarDelFinal();
            JOptionPane.showMessageDialog(null, "El elemento eliminado es: " +
el,
            "Eliminando Nodos del final", JOptionPane.INFORMATION_MESSAGE);
            break;
        case 6:
            JOptionPane.showMessageDialog(null, "Finalizado :D");
            break;
        default:
            JOptionPane.showMessageDialog(null, "Opción incorrecta");
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Error " + e.getMessage());
}
} while (opcion != 6);
}
}
```

package Simple;

```
public class Lista {
    protected Nodo inicio, fin; //Punteros para saber donde está el inicio y fin
    public Lista(){
        inicio = null;
        fin = null;
    }
}
```

```
}  
public boolean estaVacia(){  
    if(inicio == null){  
        return true;  
    } else{  
        return false;  
    }  
}  
public void agregarAlFinal(int elemento){  
    if(!estaVacia()){  
        fin.siguiete = new Nodo(elemento);  
        fin = fin.siguiete;  
    }else{  
        inicio = fin = new Nodo(elemento);  
    }  
}  
public void agregarAlInicio(int elemento){  
    inicio = new Nodo(elemento, inicio);  
    if (fin == null){  
        fin = inicio;  
    }  
}  
public void mostrarLista(){  
    Nodo recorrer = inicio;  
    System.out.println();  
    while (recorrer != null){  
        System.out.print("[“ + recorrer.dato + “]--->”);
```

```
        recorrer = recorrer.siguiete;
    }
}

public int borrarDelInicio(){
    int elemento = inicio.dato;
    if(inicio == fin){
        inicio = fin = null;
    } else{
        inicio = inicio.siguiete;
    }
    return elemento;
}

public int borrarDelFinal(){
    int elemento = fin.dato;
    if(inicio == fin){
        inicio = fin = null;
    }else{
        Nodo temporal = inicio;
        while(temporal.siguiete != fin){
            temporal = temporal.siguiete;
        }
        fin = temporal;
        fin.siguiete = null;
    }
    return elemento;
}

public void eliminar(int elemento){
    if(!estaVacia()){
```

```
if(inicio == fin && elemento == inicio.dato){
    inicio = fin = null;
}else if(elemento == inicio.dato){
    inicio = inicio.siguiete;
}else{
    Nodo anterior, temporal;

    anterior = inicio;
    temporal = inicio.siguiete;
    while(temporal != null && temporal.dato != elemento){
        anterior = anterior.siguiete;
        temporal = temporal.siguiete;
    }
    if(temporal != null){
        anterior.siguiete = temporal.siguiete;
        if(temporal == fin){
            fin = anterior;
        }
    }
}

public boolean estaEnLaLista(int elemento){
    Nodo temporal = inicio;
    while(temporal != null && temporal.dato != elemento){
        temporal = temporal.siguiete;
    }
    return temporal != null;
}
```

```
package Simple;
```

```
public class Nodo {  
    public int dato;  
    public Nodo siguiente; //Puntero enlace  
  
    // Constructor para insertar al final  
  
    public Nodo(int dato){  
        this.dato = dato;  
        this.siguiente = null;  
    }  
  
    //constructor para insertar al inicio  
  
    public Nodo(int dato, Nodo n){  
        this.dato = dato;  
        siguiente = n;  
    }  
}
```

124

Elaborar un programa en Java que elimine un elemento en específico, y que busque un elemento en la lista.

Y que use el siguiente menú:

1. Agregar número
2. Eliminar número
3. Buscar número
4. Ver lista
5. Salir

124.1 Pseint: Ejercicio propuesto

Se propone este ejercicio en Pseint. Tanto pseudocódigo como diagrama de flujo.

124.2 Adaptación a Java

```
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
public class Main {
    public static void main(String[] args) {
        List<Integer> numeros = new ArrayList<>();
        int opcion = 0;
        while (opcion != 5) {
            String menu = "Menú:\n"
                + "1. Agregar número\n"
                + "2. Eliminar número\n"
                + "3. Buscar número\n"
                + "4. Ver lista\n"
                + "5. Salir\n\n"
                + "Ingrese la opción deseada:";
        }
    }
}
```

```
opcion = Integer.parseInt(JOptionPane.showInputDialog(menu));
switch (opcion) {
    case 1:
        int numeroAgregar = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el número a agregar:"));
        numeros.add(numeroAgregar);
        JOptionPane.showMessageDialog(null, "Número agregado");
        JOptionPane.showMessageDialog(null, "Número agregado correctamente.");
        break;
    case 2:
        if (numeros.isEmpty()) {
            JOptionPane.showMessageDialog(null, "La lista está vacía. No se pueden eliminar elementos.");
        } else {
            int numeroEliminar = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el número a eliminar:"));
            if (numeros.contains(numeroEliminar)) {
                numeros.remove(Integer.valueOf(numeroEliminar));
                JOptionPane.showMessageDialog(null, "Elemento eliminado correctamente.");
            } else {
                JOptionPane.showMessageDialog(null, "El elemento no existe en la lista.");
            }
        }
        break;
    case 3:
        if (numeros.isEmpty()) {
            JOptionPane.showMessageDialog(null, "La lista está vacía. No se pueden buscar elementos.");
        } else {
```



```
int numeroBuscar = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el
número a buscar:"));

    if (numeros.contains(numeroBuscar)) {
        JOptionPane.showMessageDialog(null, "El elemento existe en la lista.");
    } else {
        JOptionPane.showMessageDialog(null, "El elemento no existe en la lista.");
    }
}
break;

case 4:
    if (numeros.isEmpty()) {
        JOptionPane.showMessageDialog(null, "La lista está vacía.");
    } else {
        StringBuilder output = new StringBuilder("Lista de números:\n");
        for (int numero : numeros) {
            output.append(numero).append("\n");
        }
        JOptionPane.showMessageDialog(null, output.toString());
    }
    break;

case 5:
    JOptionPane.showMessageDialog(null, "Saliendo del programa...");
    break;
default:
    JOptionPane.showMessageDialog(null, "Opción inválida. Intente nueva-
mente.");
    break;
}
}
}
}
```

Bloque 13: Archivos

13.1

Realice un programa que pida al usuario el nombre de un fichero de texto y, a continuación, permita almacenar al usuario tantas frases como el usuario desee.

13.1.1 Código Pseint

Algoritmo B14_Ejercicio_I

```
Definir nombre Como Cadena
Definir continuar Como Caracter
Definir archivo Como Cadena
Escribir "Ingrese un nombre:"
Leer nombre
archivo <- ""
continuar <- "s"
Mientras continuar = "s" Hacer
    Escribir "Ingrese una frase:"
    Leer frase
    archivo <- archivo + frase + " "
    Escribir "¿Desea continuar? (s/n)"
    Leer continuar
FinMientras
```

Escribir "Archivo guardado correctamente"

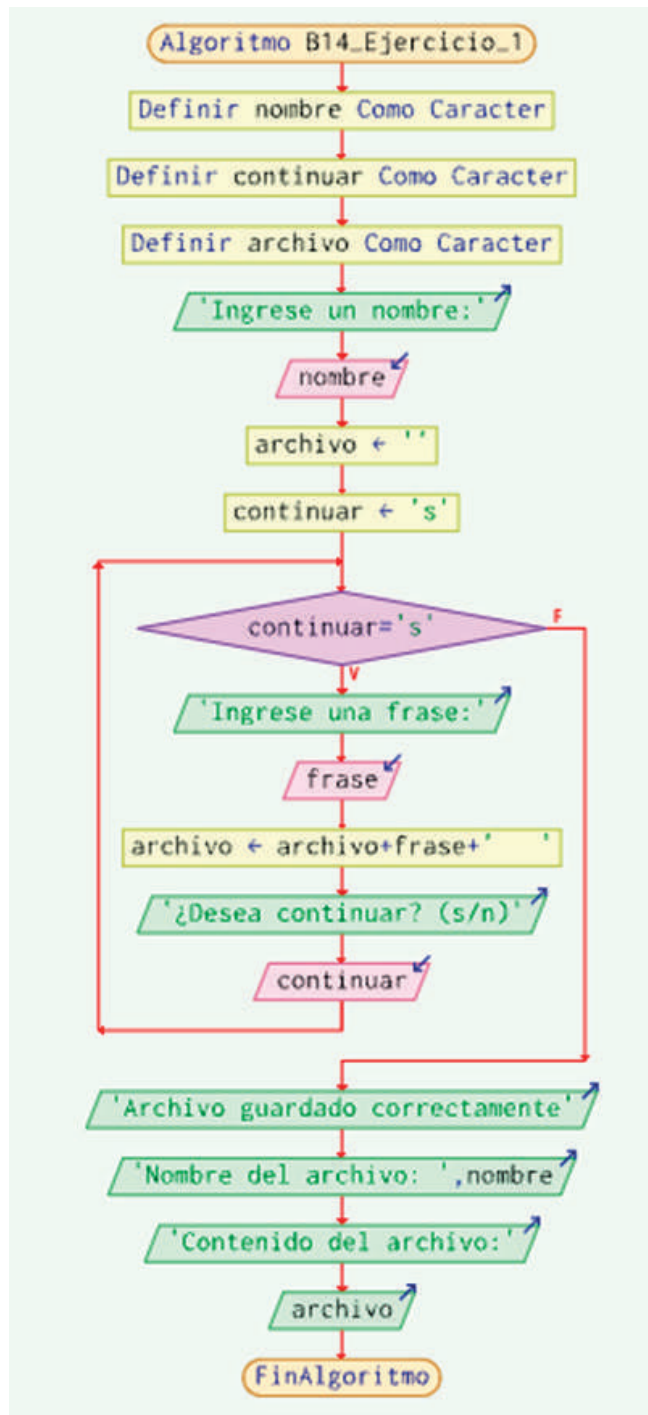
Escribir "Nombre del archivo:", nombre

Escribir "Contenido del archivo:"

Escribir archivo

FinAlgoritmo

13.1.2 Diagrama de flujo



13.1.3 Adaptación a Java

```
package Frases;
```

```
import java.io.*;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```

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

```
        String nombreArchivo, frase = "";
```

```
        boolean band = true;
```

```
        int opcion;
```

```
        File archivo;
```

```
        FileWriter escribir;
```

```
        PrintWriter linea;
```

```
        nombreArchivo = JOptionPane.showInputDialog("Nombre del fichero:");
```

```
        archivo = new File(nombreArchivo + ".txt");
```

```
        if (!archivo.exists()) {
```

```
            try {
```

```
                archivo.createNewFile();
```

```
                System.out.println("Archivo creado correctamente");
```

```
            do {
```

```
                frase = JOptionPane.showInputDialog(null, "Digite la frase:", "Solicitando frase", 3);
```

```
                escribir = new FileWriter(archivo, true);
```

```
                linea = new PrintWriter(escribir);
```

```
                linea.println(frase);
```

```
        linea.close();
        escribir.close();
        do {
            opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Continuar? (1) SI (2) NO"));
            if (opcion < 1 || opcion > 2) {
                JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
            }
        } while (opcion < 1 || opcion > 2);
        if(opcion == 2){
            band = false;
        }
    } while (band == true);
} catch (IOException ex) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
}
} else {
    try {
        do {
            frase = JOptionPane.showInputDialog(null, "Digite la frase:", "Solicitando frase", 3);
            escribir = new FileWriter(archivo, true);
            linea = new PrintWriter(escribir);
            linea.println(frase);
            linea.close();
            escribir.close();
```

```
do {
    opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Continuar? (1) SI (2) NO"));
    if (opcion < 1 || opcion > 2) {
        JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
    }
} while (opcion < 1 || opcion > 2);
if(opcion == 2){
    band = false;
}
} while (band == true);
} catch (IOException ex) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
```

13.2

Realice un programa que pida al usuario el nombre o ubicación de un fichero de texto y, a continuación de lectura a todo el fichero.

(usando el ejercicio anterior)

13.2.1 Código Pseint

Algoritmo Fichero

Definir nombre Como Cadena

Definir continuar Como Caracter

Definir archivo Como Cadena

Escribir "Ingrese un nombre:"

Leer nombre

Escribir "Ingrese el nombre o ubicación del archivo de texto:"

Leer nombreArchivo

archivo <- ""

continuar <- "s"

Mientras continuar = "s" Hacer

Escribir "Ingrese una frase:"

Leer frase

archivo <- archivo + frase + " "

Escribir "¿Desea continuar? (s/n)"

Leer continuar

FinMientras

Escribir "Archivo guardado correctamente"

Escribir "Nombre del archivo:", nombre

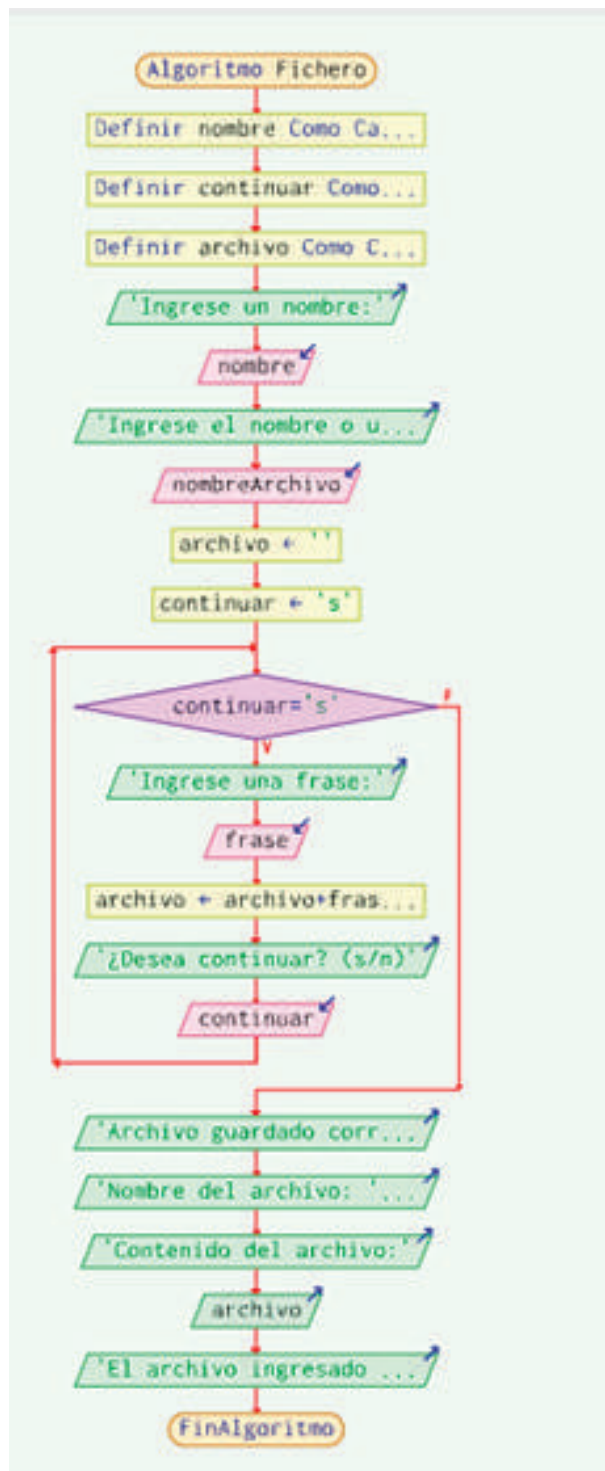
Escribir "Contenido del archivo:"

Escribir archivo

Escribir "El archivo ingresado es:", nombreArchivo

FinAlgoritmo

13.2.2 Diagrama de flujo



13.2.3 Adaptación a Java

```
import java.io.BufferedReader;
```

```
import java.io.FileReader;
```

```
import java.io.IOException;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```

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

```
        String nombreArchivo = JOptionPane.showInputDialog("Ingrese el nombre o  
ubicación del archivo de texto:");
```

```
        BufferedReader bufferedReader = null;
```

```
        try {
```

```
            bufferedReader = new BufferedReader(new FileReader(nombreArchivo));
```

```
            StringBuilder contenidoArchivo = new StringBuilder();
```

```
            String linea;
```

```
            while ((linea = bufferedReader.readLine()) != null) {
```

```
                contenidoArchivo.append(linea).append("\n");
```

```
            }
```

```
            JOptionPane.showMessageDialog(null, "Contenido del archivo:\n" + contenidoArchivo.toString());
```

```
        } catch (IOException e) {
```

```
            JOptionPane.showMessageDialog(null, "Error al acceder al archivo:" + e.getMessage());
```

```
        } finally {
```

```
            if (bufferedReader != null) {
```

```
                try {
```

```
                    bufferedReader.close();
```

```
                } catch (IOException e) {
```

```
                    JOptionPane.showMessageDialog(null, "Error al cerrar el archivo:" + e.getMessage());
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

13.3 Tercer ejercicio

Realice un programa que pida al usuario el nombre o ubicación de un fichero de texto y, a continuación, añada texto en el hasta que el usuario lo decida.

13.3.2 Código Pseint

Algoritmo B14_Ejercicio_3

Definir nombre Como Cadena

Definir continuar Como Caracter

Definir archivo Como Cadena

Definir ubicacion Como Cadena

Escribir “Elija cómo quiere buscar/crear el archivo”

Escribir “(1) por nombre”

Escribir “(2) por ubicación “

Leer opcion

Segun opcion Hacer

 I:

 Escribir “Ingrese un nombre:”

 Leer nombre

 archivo <- “”

 continuar <- “s”

 Mientras continuar = “s” Hacer

 Escribir “Ingrese una frase:”

 Leer frase

 archivo <- archivo + frase + “ “

 Escribir “¿Desea continuar? (s/n)”

 Leer continuar

FinMientras

Escribir "Archivo guardado correctamente"

Escribir "Nombre del archivo:", nombre

Escribir "Contenido del archivo:"

Escribir archivo

Escribir archivo

2:

Escribir "Ingrese ubicación:"

Leer ubicacion

Escribir "Ingrese un nombre:"

Leer nombre

archivo <- ""

continuar <- "s"

Mientras continuar = "s" Hacer

Escribir "Ingrese una frase:"

Leer frase

archivo <- archivo + frase + " "

Escribir "¿Desea continuar? (s/n)"

Leer continuar

FinMientras

Escribir "Archivo guardado correctamente"

Escribir "Ubicación:" ubicacion

Escribir "Nombre del archivo:", nombre

Escribir "Contenido del archivo:"

Escribir archivo

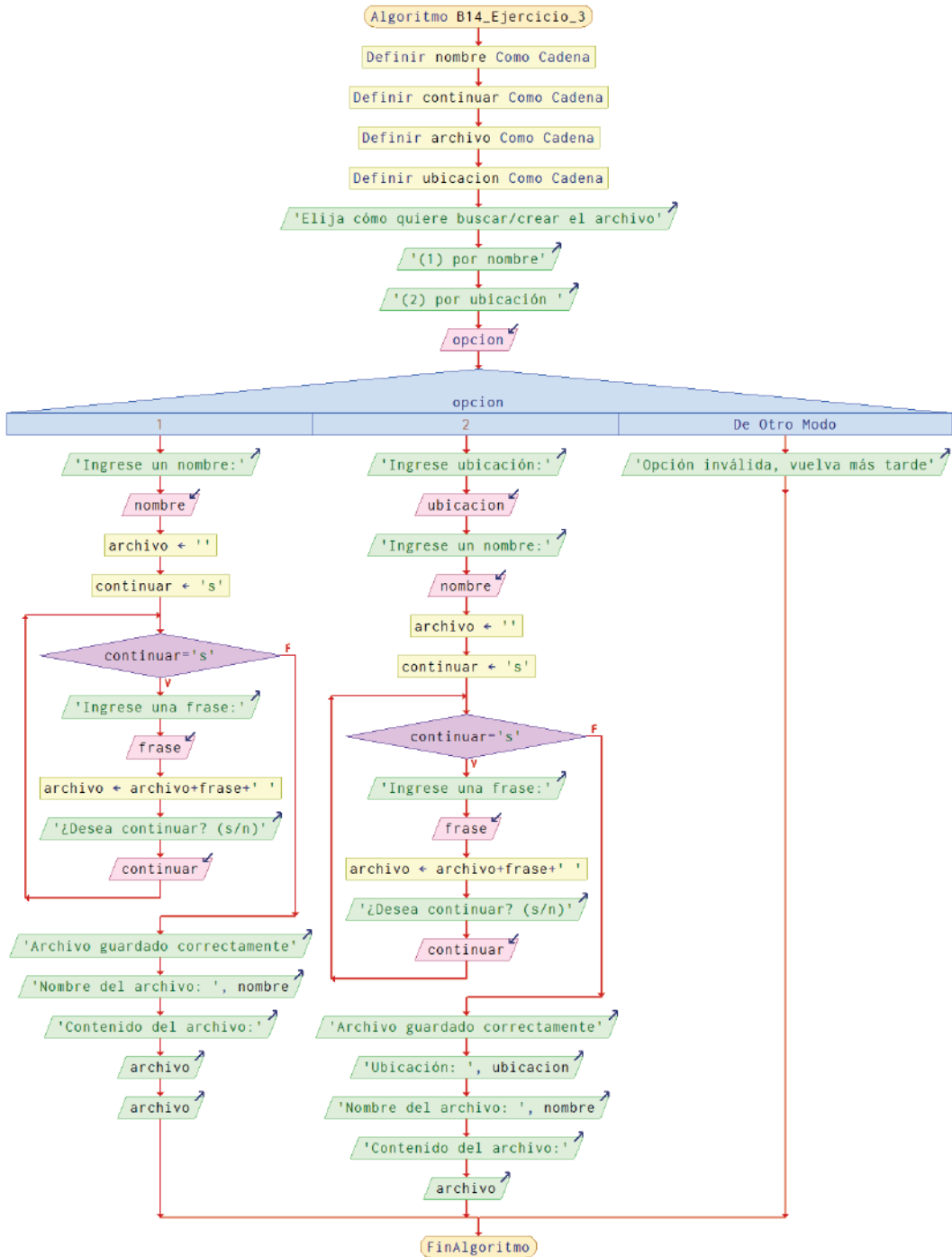
De Otro Modo:

Escribir "Opción inválida, vuelva más tarde"

Fin Segun

FinAlgoritmo

13.3.3. Diagrama de flujo



13.34 Adaptación a Java

```
package ubicacion;

import java.io.*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {

        String nombreArchivo, ubicacion = "", frase = "";

        boolean band = true;

        int opcion, opcionUbicacion;

        File archivo;

        FileWriter escribir;

        PrintWriter linea;

        do {

            opcionUbicacion = Integer.parseInt(JOptionPane.showInputDialog("Elija como quiere buscar/crear el archivo"
                + "\n (1) Por nombre"
                + "\n (2) Por ubicación"));

            if (opcionUbicacion < 1 || opcionUbicacion > 2) {
                JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
            }
        } while (opcionUbicacion < 1 || opcionUbicacion > 2);

        switch (opcionUbicacion) {

            case 1:

                System.out.println("Formato: SOLO escribir el nombre del archivo sin la extension, se guardará en la carpeta donde está instalado el programa");
                break;

            case 2:

                System.out.println("Formato: Escribir la Ubicacion completa de la carpeta donde desea guardar el archivo");
                ubicacion = JOptionPane.showInputDialog("Digite la ubicacion");
            }
        }
    }
}
```

```
ubicacion += "/";
    break;
default:

    System.out.println("No sabemos como te saltaste la verificación de seguridad, pero estamos en la búsqueda de una solución al error");
}
nombreArchivo = JOptionPane.showInputDialog("Nombre del fichero:");
archivo = new File(ubicacion + nombreArchivo + ".txt");
if (!archivo.exists()) {
    try {
        archivo.createNewFile();

        System.out.println("Archivo creado correctamente");

        do {
            frase = JOptionPane.showInputDialog(null, "Digite la frase:", "Solicitando frase", 3);

            escribir = new FileWriter(archivo, true);
            linea = new PrintWriter(escribir);
            linea.println(frase);
            linea.close();
            escribir.close();

            do {
                opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Continuar? (1) SI (2) NO"));
                if (opcion < 1 || opcion > 2) {
                    JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
                }
            } while (opcion < 1 || opcion > 2);
            if (opcion == 2) {
                band = false;
            }
        } while (band == true);
    }
}
```

```
} catch (IOException ex) {
    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
}
} else {
    try {
        do {
            frase = JOptionPane.showInputDialog(null, "Digite la frase:", "Solicitan-
do frase", 3);

            escribir = new FileWriter(archivo, true);
            linea = new PrintWriter(escribir);
            linea.println(frase);
            linea.close();
            escribir.close();
            do {
                opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Con-
tinuar? (1) SI (2) NO"));
                if (opcion < 1 || opcion > 2) {
                    JOptionPane.showMessageDialog(null, "Error, vuelva a intentar");
                }
            } while (opcion < 1 || opcion > 2);
            if (opcion == 2) {
                band = false;
            }
        } while (band == true);
    } catch (IOException ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
```


134

Hacer un programa en Java. Para guardar números telefónicos que muestre un menú con las siguientes opciones:

1. Crear (nombre, apellidos, teléfono)
2. Agregar más información.
3. Visualizar contactos existentes.

134.2 Código Pseint

SubProceso imprimirUnContacto(dato_c, dato_e, dato_r, f)

Escribir Sin Saltar "CONTACTO ", f+1, ": ";

Escribir dato_c[f,0], " -- ", dato_c[f,1], " -- ", dato_c[f,2], " -- ", dato_c[f,3], " -- ",
dato_c[f,4], " -- ", dato_c[f,5], " -- ", dato_c[f,6], " -- ", dato_e[f,0], " -- ", da-
to_e[f,1], " -- ", dato_e[f,2], " -- ", dato_r[f];

Escribir "";

FinSubProceso

////////////////////////////////////

SubProceso imprimirAgenda(dato_c, dato_e, dato_r)

Definir f como entero;

para f <- 0 hasta 19 con paso 1 Hacer

Escribir Sin Saltar "CONTACTO ", f+1, ": ";

Escribir dato_c[f,0], " -- ", dato_c[f,1], " -- ", dato_c[f,2], " -- ", dato_c[f,3], " -- ",
dato_c[f,4], " -- ", dato_c[f,5], " -- ", dato_c[f,6], " -- ", dato_e[f,0], " -- ", da-
to_e[f,1], " -- ", dato_e[f,2], " -- ", dato_r[f];

Escribir "";

FinPara

FinSubProceso

////////////////////////////////////

SubProceso menu()

Escribir “ -Bienvenido a la Agenda Telefonica-”;

Escribir “”;

Escribir “ MENU PRINCIPAL”;

Escribir “”;

Escribir “para mostrar toda la agenda telefónica digite 1”;

Escribir “Para ingresar los datos de un contacto digite 2”;

Escribir “Para ver los datos de un contacto digite 3”;

Escribir “Para modificar los datos de un contacto digite 4”;

Escribir “Para borrar los datos de un contacto digite 5”;

Escribir “Para listar los cotactos que ganan cierto salario digite 6”;

Escribir “Para listar los contactos que cumplen años en un mes concreto digite 7”;

Escribir “Para listar los que tienen mas de cierta edad digite 8”;

Escribir “Para borrar todos los contactos de la agenda digite 9”;

FinSubProceso

////////////////////////////////////

SubProceso ingresarDatos(dato_c, dato_e, dato_r, f)

Escribir “Ingrese Primer Nombre “;

Leer dato_c[f,0];

Escribir “Ingrese Segundo Nombre “;

Leer dato_c[f,1];

Escribir “Ingrese Primer Apellido “;

Leer dato_c[f,2];

Escribir “Ingrese Segundo Apellido “;

Leer dato_c[f,3];

```
Escribir "Ingrese Estado Civil ";
Leer dato_c[f,4];
Escribir "Ingrese Dirreccion de Correo Electronico ";
Leer dato_c[f,5];
Escribir "Ingrese Whatsapp si tiene ";
Leer dato_c[f,6];
Escribir "Ingrese Telefono fijo ";
Leer dato_e[f,0];
Escribir "Ingrese Numero del Celular ";
Leer dato_e[f,1];
Escribir "Digite Fecha de Nacimiento AAAAMMDD";
Leer dato_e[f,2];
Escribir "Ingrese Salario que gana ";
Leer dato_r[f];
```

FinSubProceso

SubProceso borrarDatos(dato_c, dato_e, dato_r, f)

```
dato_c[f,0] <- "xxx";
dato_c[f,1] <- "xxx";
dato_c[f,2] <- "xxx";
dato_c[f,3] <- "xxx";
dato_c[f,4] <- "xxx";
dato_c[f,5] <- "xxx";
dato_c[f,6] <- "xxx";
dato_e[f,0] <- 0;
dato_e[f,1] <- 0;
```

```

dato_e[f,2] <- 0;
dato_r[f] <- 0;
FinSubProceso
////////////////////////////////////

```

SubProceso listarSueldo(dato_c, dato_e, dato_r, salario)

```

Definir f Como Entero;
Para f<-0 Hasta 19 Con Paso 1 Hacer
  si dato_r[f] = salario Entonces
    imprimirUnContacto(dato_c, dato_e, dato_r, f);
  FinSi
FinPara
FinSubProceso
////////////////////////////////////

```

SubProceso listarCumpleMes(dato_c, dato_e, dato_r, mes)

```

Definir f Como Entero;
Definir cumple Como Caracter;
Para f<-0 Hasta 19 Con Paso 1 Hacer
  cumple <- ConvertirATexto(dato_e[f,2]);
  si SubCadena(cumple,4,5) = mes Entonces
    imprimirUnContacto(dato_c, dato_e, dato_r, f);
  FinSi
FinPara
FinSubProceso
////////////////////////////////////

```

SubProceso listarEdad(dato_c, dato_e, dato_r, edad)

```

Definir f, anio Como Entero;
Definir fecha Como Caracter;
Para f<-0 Hasta 19 Con Paso 1 Hacer
    fecha <- ConvertirATexto(dato_e[f,2]);
    anio <- ConvertirANumero(SubCadena(fecha,0,3));
    si ((2015-anio) > edad) & (anio <> 0) Entonces
        imprimirUnContacto(dato_c, dato_e, dato_r, f);
    FinSi
FinPara
FinSubProceso
////////////////////////////////////

```

SubProceso inicializaMatriz(dato_c, dato_e, dato_r)

```

Definir f, c como entero;
Para f<-0 Hasta 19 Con Paso 1 Hacer
    Para c<-0 Hasta 6 Con Paso 1 Hacer
        dato_c[f,c] <- 'xxx';
    FinPara
FinPara
Para f<-0 Hasta 19 Con Paso 1 Hacer
    Para c<-0 Hasta 2 Con Paso 1 Hacer
        dato_e[f,c] <- 0;
    FinPara
FinPara

```

```
Para f<-0 Hasta 19 Con Paso 1 Hacer
    dato_r[f] <- 0;
FinPara
FinSubProceso
Proceso Agenda_telefonica
    Definir dato_c Como Caracter;
    Definir dato_e Como Entero;
    Definir dato_r Como Real;
    Definir edad, f,c, fila, selecMenu Como entero;
    Definir R, mes como caracter;
    Definir salario Como Real;
    Dimension dato_c[20,7];
    Dimension dato_e[20,3];
    Dimension dato_r[20];
    inicializaMatriz(dato_c, dato_e, dato_r);

    R <- 's';
Mientras R = 's' | R = 'S' Hacer
    menu();
    leer selecMenu;
    Segun selecMenu Hacer
        1:
            imprimirAgenda(dato_c, dato_e, dato_r);
        2:
            Escribir "digite la fila en la que va a almacenar los datos del con-
            tacto (0-19)";
            leer fila;
            ingresarDatos(dato_c, dato_e, dato_r, fila);
        3:
            Escribir "digite la fila en la que está almacenado el contac-
            to(0-19)";
```

```
leer fila;
imprimirUnContacto(dato_c, dato_e, dato_r, fila);
4:
Escribir "digite la fila en la que está almacenado el contac-
to(0-19)";
leer fila;
ingresarDatos(dato_c, dato_e, dato_r, fila);
5:
Escribir "digite la fila en la que está almacenado el contac-
to(0-19)";
leer fila;
borrarDatos(dato_c, dato_e, dato_r, fila);
6:
Escribir "digite el salario";
leer salario;
listarSueldo(dato_c, dato_e, dato_r, salario);
7:
Escribir "digite numero del mes (01-12)";
leer mes;
listarCumpleMes(dato_c, dato_e, dato_r, mes);
8:
Escribir "digite la edad";
leer edad;

listarEdad(dato_c, dato_e, dato_r, edad);
9:
inicializaMatriz(dato_c, dato_e, dato_r);
```

De Otro Modo:

Escribir "Opción incorrecta";

FinSegun

Escribir "Desea hacer otra consulta? s/n";

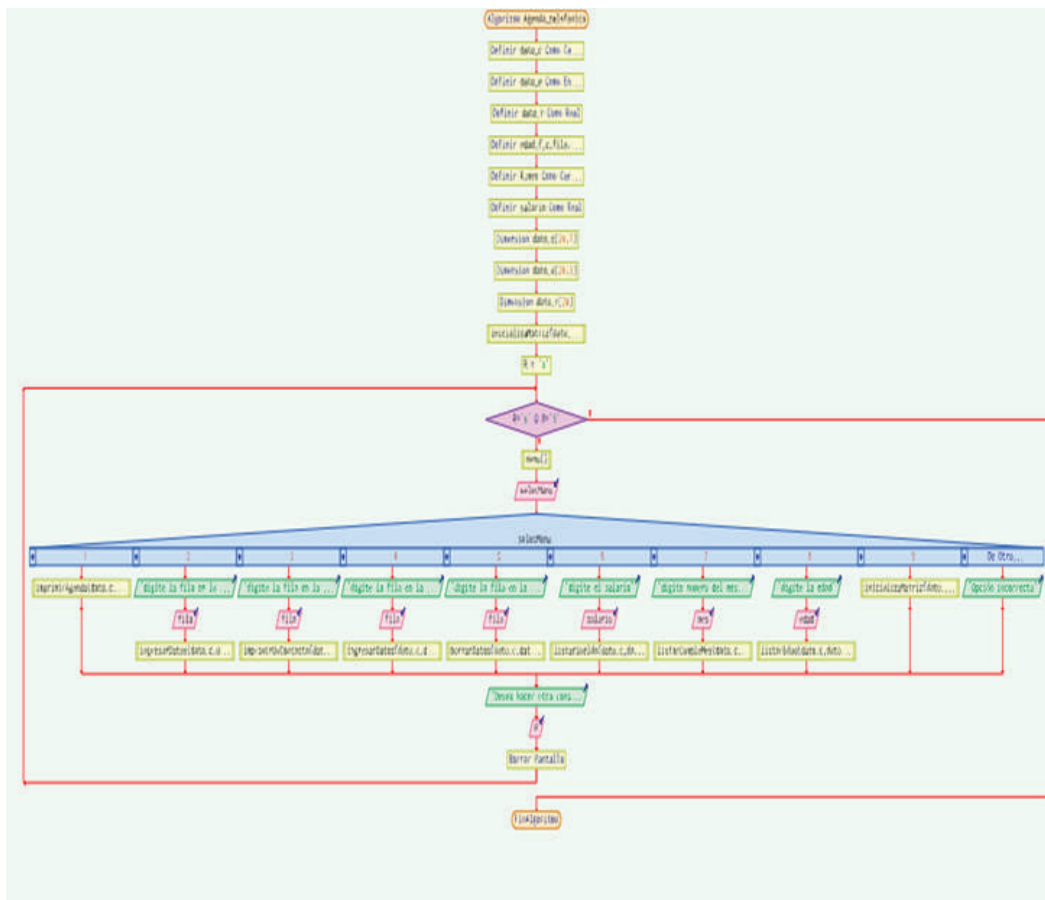
Leer R;

Borrar Pantalla;

FinMientras

FinProceso

134.3 Diagrama de flujo



1344 Adaptación a Java

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import javax.swing.JOptionPane;

public class Main {
    private static final String NOMBRE_ARCHIVO = "agenda.txt";
    public static void main(String[] args) {
        Map<String, Contacto> agenda = new HashMap<>();
        int opcion;
        do {
            String menu = "Agenda Telefónica\n\n"
                + "1. Crear contacto\n"
                + "2. Agregar más Información\n"
                + "3. Visualizar contactos existentes\n"
                + "4. Salir\n\n"
                + "Ingrese la opción deseada:";
            opcion = Integer.parseInt(JOptionPane.showInputDialog(null, menu));
            switch (opcion) {
                case 1:
                    crearContacto(agenda);
                case 2:
                    crearContacto(agenda);
                break;
            }
        }
    }
}
```

```
        case 2:
            agregarContacto(agenda);

            break;

        case 3:
            visualizarContactos(agenda);

            break;

        case 4:
            guardarAgenda(agenda);

            JOptionPane.showMessageDialog(null, "Saliendo del programa...");

            break;

        default:
            JOptionPane.showMessageDialog(null, "Opción inválida. Intente nuevamente.");
    }
} while (opcion != 4);
}

public static void crearContacto(Map<String, Contacto> agenda) {
    String nombre = JOptionPane.showInputDialog("Ingrese el nombre:");
    String apellidos = JOptionPane.showInputDialog("Ingrese los apellidos:");
    String telefono = JOptionPane.showInputDialog("Ingrese el número de teléfono:");

    Contacto contacto = new Contacto(nombre, apellidos, telefono);
    agenda.put(nombre, contacto);

    JOptionPane.showMessageDialog(null, "Contacto creado correctamente.");
}

public static void agregarContacto(Map<String, Contacto> agenda) {
    String nombre = JOptionPane.showInputDialog("Ingrese el nombre del contacto a agregar más información:");
```

```
    Contacto contacto = agenda.get(nombre);
    if (contacto != null) {
        String informacionAdicional = JOptionPane.showInputDialog("Ingrese la
información adicional:");
        contacto.agregarInformacionAdicional(informacionAdicional);
        JOptionPane.showMessageDialog(null, "Información adicional agregada co-
rrectamente.");
    } else {
        JOptionPane.showMessageDialog(null, "No se encontró un contacto con ese
nombre.");
    }
}

public static void visualizarContactos(Map<String, Contacto> agenda) {
    if (agenda.isEmpty()) {
        JOptionPane.showMessageDialog(null, "No hay contactos almacenados en la
agenda.");
    } else {
        StringBuilder output = new StringBuilder("Contactos existentes:\n\n");
        for (Contacto contacto : agenda.values()) {
            output.append(contacto).append("\n");
        }
        JOptionPane.showMessageDialog(null, output.toString());
    }
}

public static void guardarAgenda(Map<String, Contacto> agenda) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(NOMBRE_
ARCHIVO))) {
        for (Contacto contacto : agenda.values()) {
            writer.write(contacto.toString());
        }
    }
}
```

```
public static void guardarAgenda(Map<String, Contacto> agenda) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(NOMBRE_
        ARCHIVO))) {
        for (Contacto contacto : agenda.values()) {
            writer.write(contacto.toString());
            writer.newLine();
        }
        JOptionPane.showMessageDialog(null, "Agenda guardada correctamente en "
            + NOMBRE_ARCHIVO);
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "Error al guardar la agenda:" + e.get-
            Message());
    }
}

class Contacto {
    private String nombre;
    private String apellidos;
    private String telefono;
    private StringBuilder informacionAdicional;
    public Contacto(String nombre, String apellidos, String telefono) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.telefono = telefono;
        this.informacionAdicional = new StringBuilder();
    }
}
```

```
public void agregarInformacionAdicional(String informacion) {
    informacionAdicional.append(informacion).append("\n");
public void agregarInformacionAdicional(String informacion) {
    informacionAdicional.append(informacion).append("\n");
}
@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("Nombre:").append(nombre).append("\n");
    sb.append("Apellidos:").append(apellidos).append("\n");
    sb.append("Teléfono:").append(telefono).append("\n");
    sb.append("Información adicional: \n").append(informacionAdicional);
    return sb.toString();
}
}
```

Bloque 14: Programación Orientada a Objetos (POO)

14.1

Construya una clase llamada Rectángulo que tenga los siguientes atributos: largo y ancho, y los siguientes métodos: perímetro() y área()

14.1.1 Código Pseint

SubProceso CalcularPerimetro(largo, ancho)

Definir perimetro Como Real

perimetro $\leftarrow 2 * (\text{largo} + \text{ancho})$

Escribir “El perímetro del rectángulo es:”, perimetro

FinSubProceso

SubProceso CalcularArea(largo, ancho)

Definir area Como Real

area $\leftarrow \text{largo} * \text{ancho}$

Escribir “El área del rectángulo es:”, area

FinSubProceso

Algoritmo B15_Ejercicio_1

Definir largo, ancho Como Real

Definir opción Como Caracter

Escribir "Ingrese el largo del rectángulo:"

Leer largo

Escribir "Ingrese el ancho del rectángulo:"

Leer ancho

Escribir "Elija una opción:"

Escribir "a) Calcular perímetro"

Escribir "b) Calcular área"

Escribir "Ingrese la opción elegida (a/b):"

Leer opción

Si opción = "a" Entonces

 CalcularPerimetro(largo, ancho)

Sino Si opción = "b" Entonces

 CalcularArea(largo, ancho)

Sino

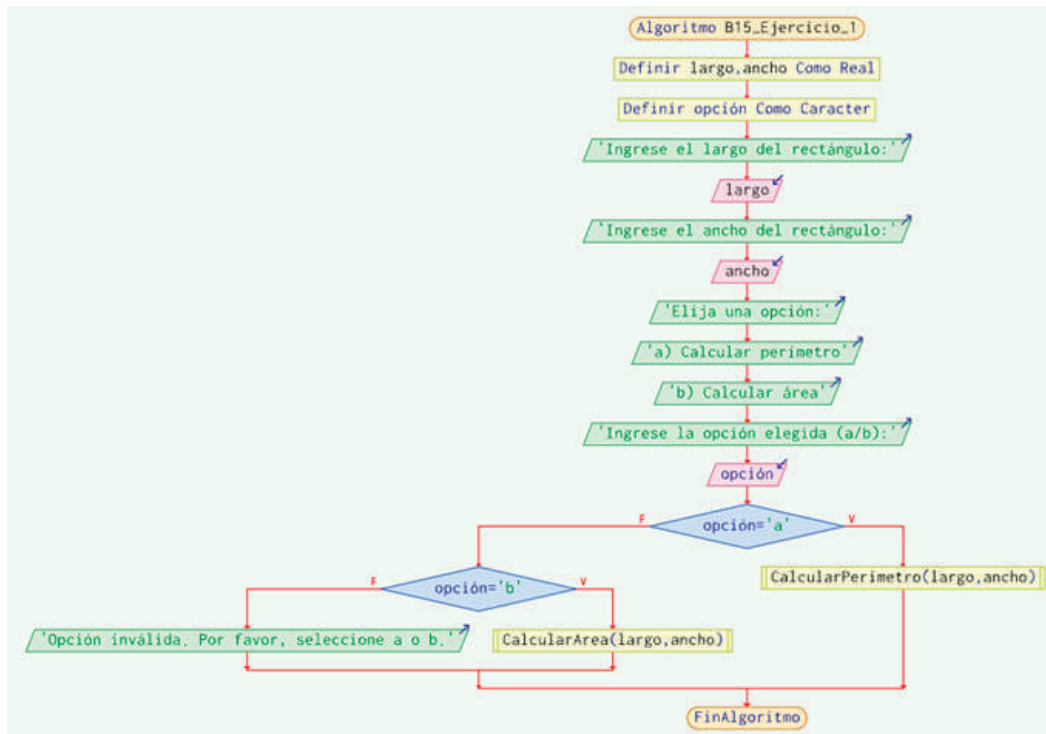
 Escribir "Opción inválida. Por favor, seleccione a o b."

FinSi

FinSi

FinAlgoritmo

14.1.2 Diagrama de flujo



14.1.3 Adaptación a Java

package poo;

import javax.swing.JOptionPane;

public class Main {

public static void main(String[] args){

int total;

double largo, ancho;

total = Integer.parseInt(JOptionPane.showInputDialog(null,"Digite el total de rectángulos"));

for(int i = 0; i<total; i++){

largo = Integer.parseInt(JOptionPane.showInputDialog(null,"Digite el largo"));


```
        ancho = Integer.parseInt(JOptionPane.showInputDialog(null,"Digite el ancho"));
    Rectángulo figura = new Rectángulo(largo, ancho);
    System.out.println("Perimetro del rectangulo ["++]:" + figura.perimetro());
    System.out.println("Area del rectangulo ["++]:" + figura.area());
    System.out.println();
    }
}
package poo;
public class Rectángulo {
    private double largo;
    private double ancho;
    public Rectángulo(double largo, double ancho) {
        this.largo = largo;
        this.ancho = ancho;
    }
    public double perimetro(){
        double perimetro = 2*largo + 2*ancho;
        return perimetro;
    }
    public double area(){
        double area = largo * ancho;
        return area;
    }
}
```

14.2

Construya una clase `Tiempo` que contenga los siguientes atributos enteros: horas, minutos y segundos. Haga que la clase contenga 2 constructores, el primero debe tener 3 parámetros `Tiempo(int,int,int)` y el segundo sólo tendrá un campo que serán los segundos y desensamble el número entero largo en horas, minutos y segundos.

14.2.1 Adaptación a Java

```
public class Tiempo {  
    private int horas;  
    private int minutos;  
    private int segundos;  
    public Tiempo(int horas, int minutos, int segundos) {  
        this.horas = horas;  
        this.minutos = minutos;  
        this.segundos = segundos;  
    }  
    public Tiempo(int totalSegundos) {  
        desensamblarSegundos(totalSegundos);  
    }  
    public int getHoras() {  
        return horas;  
    }  
    public void setHoras(int horas) {  
        this.horas = horas;  
    }  
    public int getMinutos() {  
        return minutos;  
    }  
    public void setMinutos(int minutos) {  
        this.minutos = minutos;  
    }  
}
```

```
public int getSegundos() {
    return segundos;
}
public void setSegundos(int segundos) {
    this.segundos = segundos;
}

private void desensamblarSegundos(int totalSegundos) {
    horas = totalSegundos / 3600;
    minutos = (totalSegundos % 3600) / 60;
    segundos = totalSegundos % 60;
}
}
//Puedes utilizar esta clase en tu programa para crear objetos Tiempo y acceder a
sus atributos. Aquí tienes un ejemplo:

public class Main {

    public static void main(String[] args) {
        Tiempo tiempo1 = new Tiempo(2, 30, 45);
        int horas1 = tiempo1.getHoras();
        int minutos1 = tiempo1.getMinutos();
        int segundos1 = tiempo1.getSegundos();

        System.out.println("Tiempo 1: " + horas1 + "h " + minutos1 + "m " + segundos1 + "s");

        Tiempo tiempo2 = new Tiempo(10000);
        int horas2 = tiempo2.getHoras();
        int minutos2 = tiempo2.getMinutos();
        int segundos2 = tiempo2.getSegundos();

        System.out.println("Tiempo 2: " + horas2 + "h " + minutos2 + "m " + segundos2 + "s");
    }
}
```

14.3

Realice un programa en JAVA, de tal manera que se construya una solución para la jerarquía(herencia) de clases mostrada en la siguiente figura:

14.1 Código Pseint

Algoritmo B15_Ejercicio_3

Definir nombre, direccion Como Cadena

Definir salario, matricula Como Real

Escribir "Ingrese los datos de la persona:"

Escribir "Nombre:"

Leer nombre

Escribir "Dirección:"

Leer direccion

Escribir "Ingrese los datos del empleado:"

Escribir "Salario:"

Leer salario

Escribir "Ingrese los datos del estudiante:"

Escribir "Matrícula:"

Leer matricula

Escribir "Ingrese los datos del universitario:"

Escribir "Carrera:"

Leer carrera

Escribir ""

Escribir "Datos de la persona:"

Escribir "Nombre:", nombre

Escribir “Dirección:”, direccion

Escribir “”

Escribir “Datos del empleado:”

Escribir “Salario:”, salario

Escribir “”

Escribir “Datos del estudiante:”

Escribir “Matrícula:”, matricula

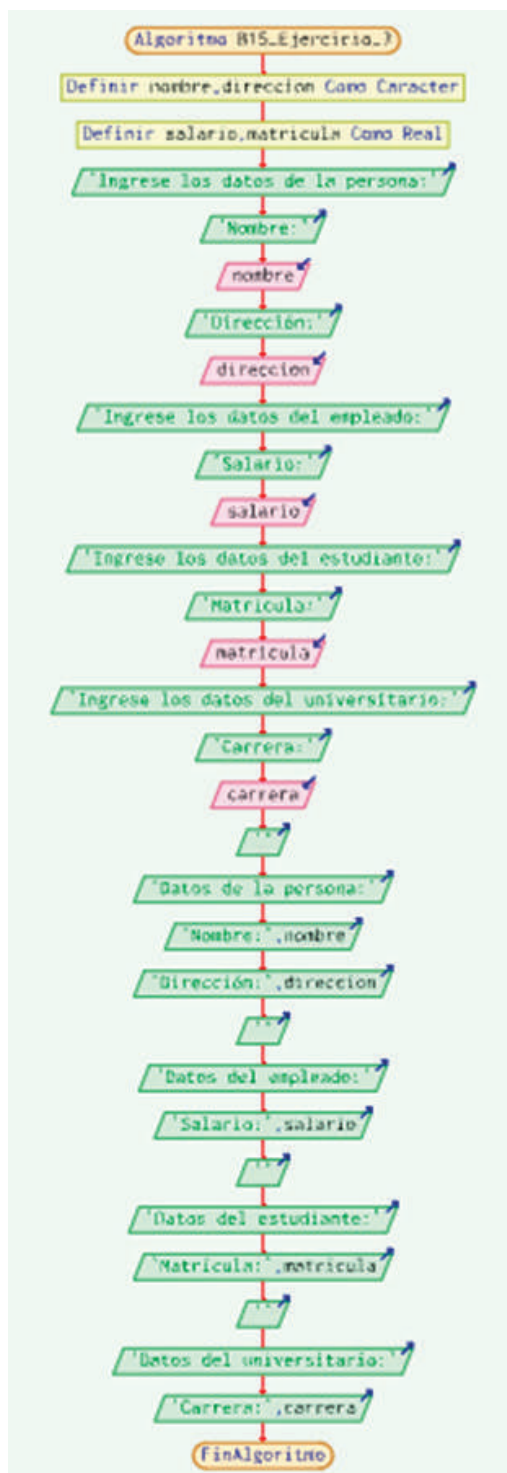
Escribir “”

Escribir “Datos del universitario:”

Escribir “Carrera:”, carrera

FinAlgoritmo

14.2 Diagrama de flujo



144.3 Adaptación a Java

```
package Herencia;
```

```
public class Persona {  
    private String nombre;  
    private String apellido;  
    private int edad;  
    public Persona(String nombre, String apellido, int edad) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.edad = edad;  
    }  
    public String getNombre() {  
        return nombre;  
    }  
    public String getApellido() {  
        return apellido;  
    }  
    public int getEdad() {  
        return edad;  
    }  
    public void mostrarDatos(){  
        System.out.println("Nombre:" + getNombre()  
            +"\nApellido:" + getApellido()  
            +"\nEdad:" + getEdad());  
    }  
}
```

package Herencia;

```
public class Universitario extends Estudiante{
    private String nombreUniversidad;

    public Universitario(String nombre, String apellido, int edad, int codigo, String marcaCuaderno, String nombreUniversidad){
        super(nombre, apellido, edad, codigo, marcaCuaderno);
        this.nombreUniversidad = nombreUniversidad;
    }
    @Override
    public void mostrarDatos(){
        System.out.println("Nombre:" + getNombre()
            +"\nApellido:" + getApellido()
            +"\nEdad:" + getEdad()
            +"\nCodigo:" + getCodigo()
            +"\nMarca del cuaderno:" + getMarcaCuaderno()
            +"\nNombre de la universidad:" + nombreUniversidad);
    }
}
```

package Herencia;

```
public class Estudiante extends Persona{
    private int codigo;
    private String marcaCuaderno;

    public Estudiante(String nombre, String apellido, int edad, int codigo, String marcaCuaderno){
        super(nombre, apellido, edad);
        this.codigo = codigo;
        this.marcaCuaderno = marcaCuaderno;
    }
}
```



```
public int getCodigo() {
    return codigo;
}

public String getMarcaCuaderno() {
    return marcaCuaderno;
}

@Override
public void mostrarDatos(){
    System.out.println("Nombre:" + getNombre()
        +"\nApellido:" + getApellido()
        +"\nEdad:" + getEdad()
        +"\nCodigo:" + codigo
        +"\nMarca del cuaderno:" +marcaCuaderno);
}
}
```

package Herencia;

```
import javax.swing.JOptionPane;

public class Main {
    public static void main(String[] args){
        int opcion, codigo, edad, totalHorasTrabajo, universitario;
        String nombre, apellido, nombreUniversidad, empresa, marcaCuaderno;
        nombre = JOptionPane.showInputDialog(null, "Digite su nombre");
        apellido = JOptionPane.showInputDialog(null, "Digite su apellido");
        edad = Integer.parseInt(JOptionPane.showInputDialog(null, "Digite su edad"));
        opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "Usted es: \n(1)
Empleado\n(2) Estudiante"));
        switch (opcion){
```

```
case 1:
    empresa = JOptionPane.showInputDialog(null, "Digite su empresa");
    totalHorasTrabajo = Integer.parseInt(JOptionPane.showInputDialog(null,
"Digite cuantas horas trabaja"));
    Empleado empleado = new Empleado(nombre, apellido, edad, empresa,
totalHorasTrabajo);
    empleado.mostrarDatos();
    break;

case 2:
    codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "Digite su
código de estudiante"));
    marcaCuaderno = JOptionPane.showInputDialog(null, "Digite la marca de
su cuaderno");
    universitario = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Es
universitario?\n(1) SI\n(2) NO"));
    switch (universitario){
        case 1:
            nombreUniversidad = JOptionPane.showInputDialog(null, "Digite el
nombre de su Universidad");
            Universitario cachimbo = new Universitario(nombre, apellido, edad,
codigo, marcaCuaderno, nombreUniversidad);
            cachimbo.mostrarDatos();
            break;
        case 2:
            Estudiante estudiante = new Estudiante(nombre, apellido, edad, codi-
go, marcaCuaderno);
            estudiante.mostrarDatos();
            break;
    }
    break;
}
```

```
// Reto propuesto, ¿Qué pasaría si la persona no es Estudiante ni empleado?  
    Cómo y qué datos mostraría  
}  
  
package Herencia;  
  
public class Empleado extends Persona{  
  
    private String empresa;  
  
    private int totalHorasTrabajo;  
  
    public Empleado(String nombre, String apellido, int edad, String empresa, int total-  
        HorasTrabajo){  
  
        super(nombre, apellido, edad);  
  
        this.empresa = empresa;  
  
        this.totalHorasTrabajo = totalHorasTrabajo;  
  
    }  
  
    @Override  
  
    public void mostrarDatos(){  
  
        System.out.println("Nombre:" + getNombre()  
            +"\nApellido:" + getApellido()  
            +"\nEdad:" + getEdad()  
            +"\nEmpresa:" + empresa  
            +"\nHoras de trabajo:" +totalHorasTrabajo);  
  
    }  
  
}
```

144

Crear un programa en Java que tenga la siguiente jerarquía de clases: Animal (Clase Padre) -> Humano (Clase Hija) -> Perro (Clase Hija), y hacer polimorfismo con el método comer().

144.1 Pseint: Ejercicio propuesto

Se propone este ejercicio para ser desarrollado en Pseint.

144.2 Adaptación a Java

```
class Animal {
    public void comer() {
        System.out.println("El animal está comiendo.");
    }
}

class Humano extends Animal {
    @Override
    public void comer() {
        System.out.println("El humano está comiendo con cubiertos.");
    }
}

class Perro extends Animal {
    @Override
    public void comer() {
        System.out.println("El perro está comiendo con entusiasmo.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal animal1 = new Animal();
        Animal animal2 = new Humano();
        Animal animal3 = new Perro();
        animal1.comer();
        animal2.comer();
        animal3.comer();
    }
}
```

14.5

En esta aplicación, representa la venta de boletos en un cine, cada función con un precio diferente, primero creamos las clases Funcion y VentaEntradas. La clase Funcion representa una función de cine con un nombre y un precio. La clase VentaEntradas gestiona la venta de entradas y tiene una lista de funciones disponibles. El método mostrarMenuFunciones() muestra las funciones disponibles con sus respectivos precios, y el método venderEntradas() permite al usuario seleccionar una función y la cantidad de entradas que desea comprar. Luego, muestra un resumen del detalle de la compra, incluyendo la función seleccionada, el precio por entrada, la cantidad de entradas y el total a pagar.

En el método main, creamos una instancia de VentaEntradas, mostramos el menú de funciones disponibles y realizamos una venta de entradas de ejemplo.

14.5.1 Código Pseint

Algoritmo VentaBoletosCine

Definir total_boletos, total_venta, precio_boleto Como Real

Definir continuar, seguir_comprando Como Caracter

total_boletos <- 0

total_venta <- 0

precio_boleto <- 100

continuar <- "S"

Mientras continuar = "S" Hacer

 Escribir "Venta de Boletos en el Cine"

 Escribir "-----"

 Escribir ""

 Escribir "Ingrese la cantidad de boletos a comprar:"

 Leer boletos_comprar

 total_boletos <- total_boletos + boletos_comprar

 total_venta <- total_venta + (boletos_comprar * precio_boleto)

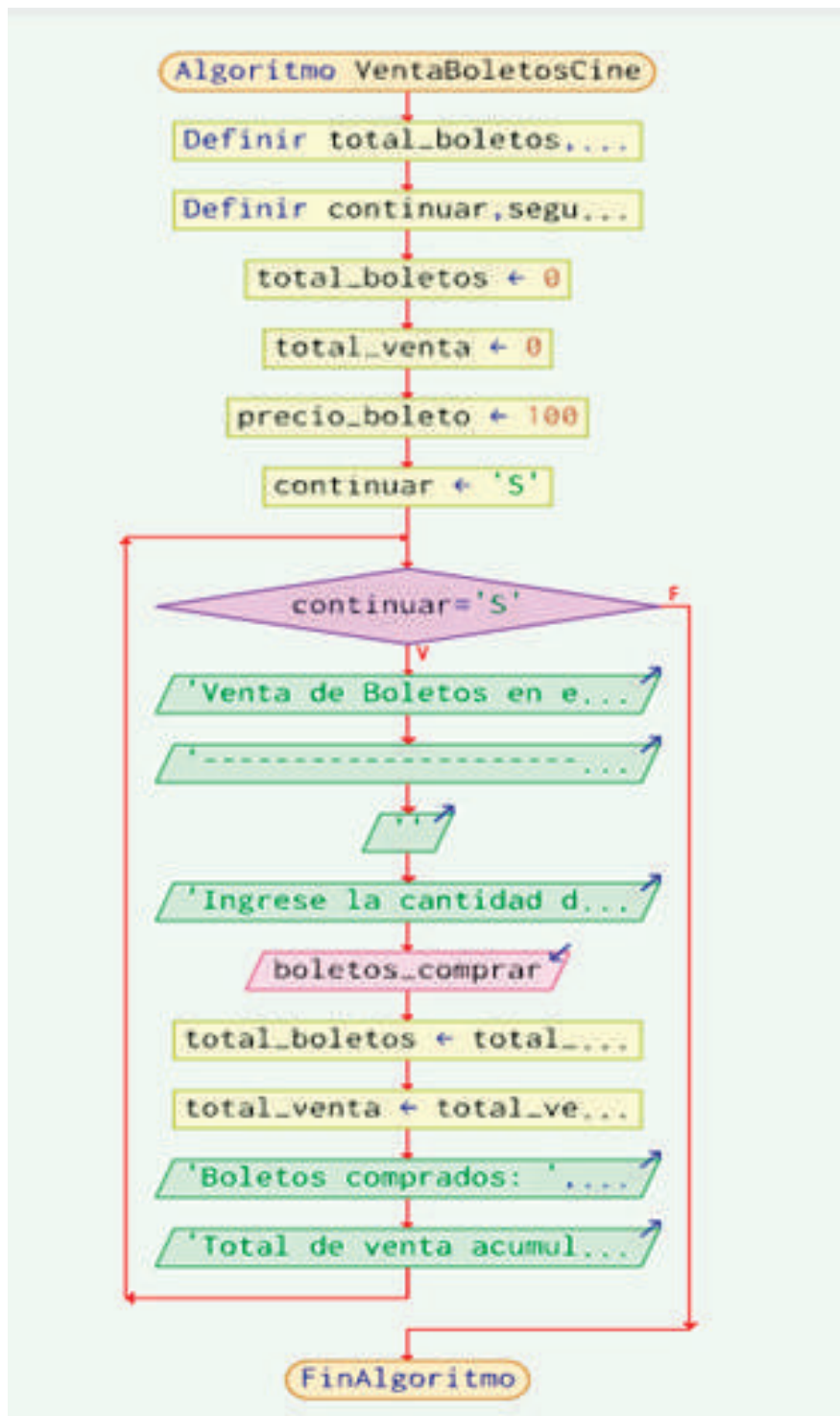
 Escribir "Boletos comprados:", boletos_comprar

 Escribir "Total de venta acumulado: \$", total_venta

 FinMientras

FinAlgoritmo

14.5.2 Diagrama de flujo



14.5.3 Adaptación a Java

```
package EstoSiEsCine;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```

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

```
        Funcion funcion = new Funcion();
```

```
        VentaEntradas venta = new VentaEntradas();
```

```
        JOptionPane.showMessageDialog(null, "Bienvenido al cine de la FISI");
```

```
        funcion.mostrarMenuFunciones();
```

```
        venta.venderEntradas();
```

```
    }
```

```
}
```

```
package EstoSiEsCine;
```

```
import javax.swing.JOptionPane;
```

```
public class VentaEntradas {
```

```
    String nombrePelicula;
```

```
    int opcion, entradas, precio;
```

```
    public void venderEntradas(){
```

```
        opcion = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Para qué película desea comprar boletos?"
```

```
        + "\n1. Mario Bros, The Movie"+
```

```
        "\n2. Spider-Man: A través del Spider-Verso"+
```

```
        "\n3. Black Clover: La espada del rey mago"+
```

```
        "\n4. Transformers: El despertar de las bestias"
```

```
        + "\n5. No me interesa ninguna película"));
```

```
        switch(opcion){
```

```
case 1:
    nombrePelicula = "Mario Bros, The Movie";

    precio = 10;

    break;

case 2:
    nombrePelicula = "Spider-Man: A través del Spider-Verso";

    precio = 12;

    break;

case 3:
    nombrePelicula = "Black Clover: La espada del rey mago";

    precio = 10;

    break;

case 4:
    nombrePelicula = "Transformers: El despertar de las bestias";

    precio = 12;

    break;

default:
    nombrePelicula = "No pidió boletos";
}

if(opcion != 5){
    entradas = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Cuántas
entradas desea comprar?"));

    System.out.println("Resumen:");

    System.out.println("Función seleccionada: " + nombrePelicula
        + "\nPrecio por entrada: " + precio
        + "\nCantidad de entradas: " + entradas
        + "\nTotal a pagar: " + (precio*entradas));
```



```
        } else{
            System.out.println(nombrePelicula);
        }
    }
}

package EstoSiEsCine;
import javax.swing.JOptionPane;
public class Funcion {
    public void mostrarMenuFunciones(){
        JOptionPane.showMessageDialog(null,“FUNCIONES DISPONIBLES\n1. Mario
        Bros,The Movie. 10 soles”+
            “\n2. Spider-Man:A través del Spider-Verso. 12 soles”+
            “\n3. Black Clover: La espada del rey mago. 10 soles”+
            “\n4.Transformers: El despertar de las bestias. 12 soles”);
    }
}
}
```

14.6

En esta aplicación, se crea una factura para una lavandería creamos las clases Item y Factura. La clase Item representa un elemento individual de la factura, con un nombre y un precio. La clase Factura gestiona una lista de ítems y proporciona métodos para calcular el subtotal, el impuesto y el total de la factura. También tiene un método para imprimir la factura en la consola.

En el método main, creamos una instancia de Factura, agregamos algunos ítems (representando servicios de lavandería) y luego llamamos al método imprimirFactura() para mostrar el resultado en la consola.

14.6.1 Código Pseint

Algoritmo factura

```
Dimension prendas[100], precios[100]
Definir nombre_cliente, fecha Como Cadena
Definir subtotal, descuento, total Como Real
Definir cantidad_prendas, descuento_porcentaje Como Entero
Escribir "Factura Lavandería"
Escribir "Ingrese el nombre del cliente:"
Leer nombre_cliente
Escribir "Ingrese la fecha de la factura:"
Leer fecha
Escribir "Ingrese la cantidad de prendas:"
Leer cantidad_prendas
Para i <- 1 Hasta cantidad_prendas Hacer
    Escribir "Ingrese la descripción de la prenda ", i, ":"
    Leer prendas[i]
    Escribir "Ingrese el precio de la prenda ", i, ":"
    Leer precios[i]
```

```

FinPara
subtotal <- 0
subtotal <- 0

Para i <- 1 Hasta cantidad_prendas Hacer
    subtotal <- subtotal + precios[i]
FinPara

Escribir "Ingrese el descuento (%):"
Leer descuento_porcentaje
descuento <- subtotal * (descuento_porcentaje / 100)
total <- subtotal - descuento

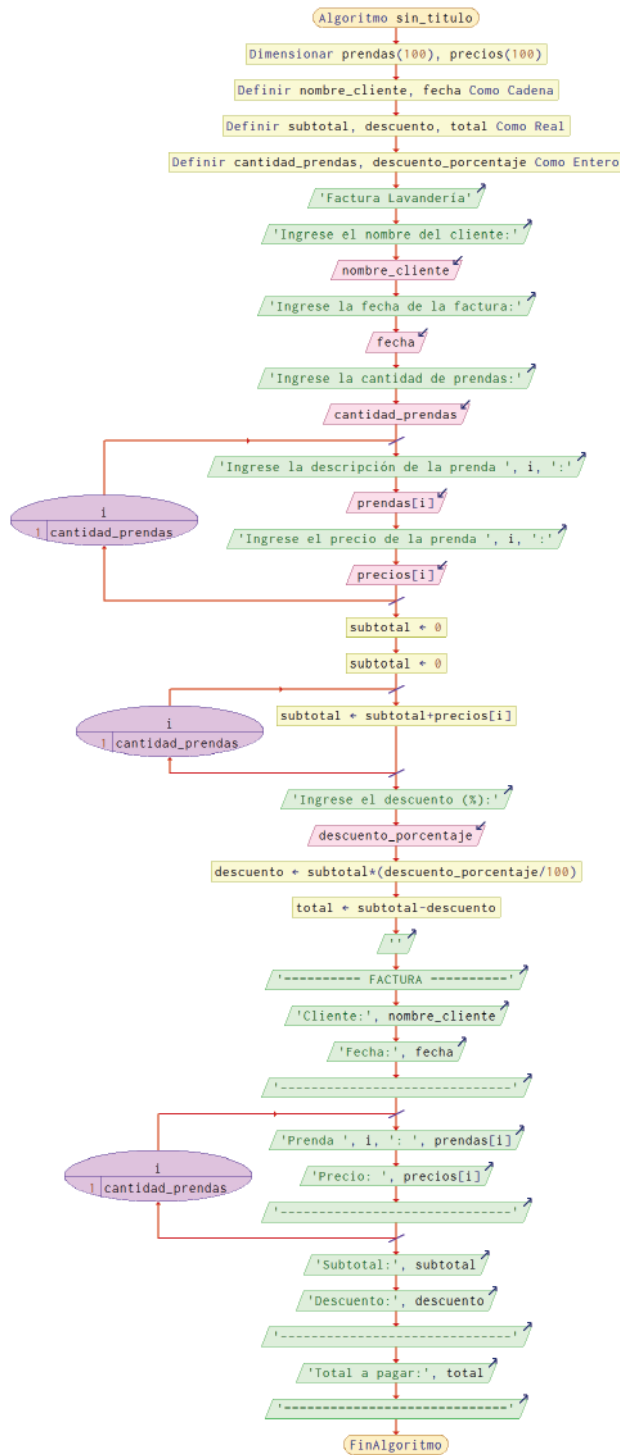
Escribir ""
Escribir "===== FACTURA ====="
Escribir "Cliente:", nombre_cliente
Escribir "Fecha:", fecha
Escribir "-----"

Para i <- 1 Hasta cantidad_prendas Hacer
    Escribir "Prenda ", i, ": ", prendas[i]
    Escribir "Precio:", precios[i]
    Escribir "-----"
FinPara

Escribir "Subtotal:", subtotal
Escribir "Descuento:", descuento
Escribir "-----"
Escribir "Total a pagar:", total
Escribir "=====
```

FinAlgoritmo

14.6.2 Diagrama de Flujo



14.6.3 Adaptación a Java

```
import javax.swing.JOptionPane;
import java.util.ArrayList;
import java.util.List;
public class Main {
    public static void main(String[] args) {
        Factura factura = new Factura();
        factura.agregarItem(new Item("Lavado de ropa", 25.0));
        factura.agregarItem(new Item("Secado de ropa", 15.0));
        factura.imprimirFactura();
    }
}
class Item {
    private String nombre;
    private double precio;
    public Item(String nombre, double precio) {
        this.nombre = nombre;
        this.precio = precio;
    }
    public double getPrecio() {
        return precio;
    }
    public String toString() {
        return nombre + ": $" + precio;
    }
}
```

```
    }  
}  
class Factura {  
    private List<Item> items;  
    public Factura() {  
        items = new ArrayList<>();  
    }  
    public void agregarItem(Item item) {  
        items.add(item);  
    }  
    public double calcularSubtotal() {  
        double subtotal = 0;  
        for (Item item : items) {  
            subtotal += item.getPrecio();  
        }  
        return subtotal;  
    }  
    public double calcularImpuesto() {  
        return calcularSubtotal() * 0.16; // Suponemos un impuesto del 16%  
    }  
    public double calcularTotal() {  
        return calcularSubtotal() + calcularImpuesto();  
    }  
}
```

```
public void imprimirFactura() {
    StringBuilder factura = new StringBuilder();
    factura.append("=== FACTURA ===\n");
    for (Item item : items) {
        factura.append(item.toString()).append("\n");
    }
    factura.append("\nSubtotal: $").append(calcularSubtotal());
    factura.append("\nImpuesto: $").append(calcularImpuesto());
    factura.append("\nTotal: $").append(calcularTotal());
    JOptionPane.showMessageDialog(null, factura.toString(), "Factura", JOptionPane.INFORMATION_MESSAGE);
}
}
```




Dr. Juan Carlos García Castro

Ing. Mg. Dr. Ing. de Sistemas y Master of Science in Engineering por The Lvov Politecnical Institute, URSS. Magister en Dirección de tecnologías de la Información Universidad ESAN, Doctor en Gestión Pública y Gobernabilidad – Cesar Vallejo y otros grados académicos.

Amplia experiencia en Dirección de Negocios y consultoría.

Dr. Jorge Damian Valverde Iparraguirre

Ingeniero de Computación y Sistemas, egresado de la Universidad Privada Antenor Orrego de Trujillo, docente en la Universidad Nacional de San Martín con más de 20 años de experiencia enseñando los cursos de algoritmos, programación y software.



Dr. Ángel Cárdenas García

Ingeniero de Sistemas, cuenta con un Doctorado en Gestión Empresarial, una Maestría en Administración de Negocios y otra en Ciencias con mención en TICs. Docente universitario con más de 15 años de experiencia en pre y postgrado.

Dr. Wilson Torres Delgado

Licenciado en Estadística, Maestro en Ciencias Económicas y Doctor en Ciencias Educativas. Docente Universitario en pre y postgrado, en las áreas de ciencias matemáticas, estadística, metodología de la investigación científica.

Actualmente docente a tiempo completo en la Universidad nacional de San Martín.

