

# ALGORITMOS

CURSO: SOFTWARE

ALUMNO: ARIEL CALDERÓN CUEVA

DOCENTE: MSc. ENRIQUE BAÑO LEÓN

## Problema: Calcular el perímetro de un rectángulo

Diseña un programa que solicite al usuario ingresar la longitud y el ancho de un rectángulo. Asegúrate de validar que ambos valores sean números positivos. Si el usuario ingresa un valor no válido, muestra un mensaje de error. Si los valores son válidos, calcula el perímetro del rectángulo utilizando la fórmula  $2 \times (\text{longitud} + \text{ancho})$  y muestra el resultado.

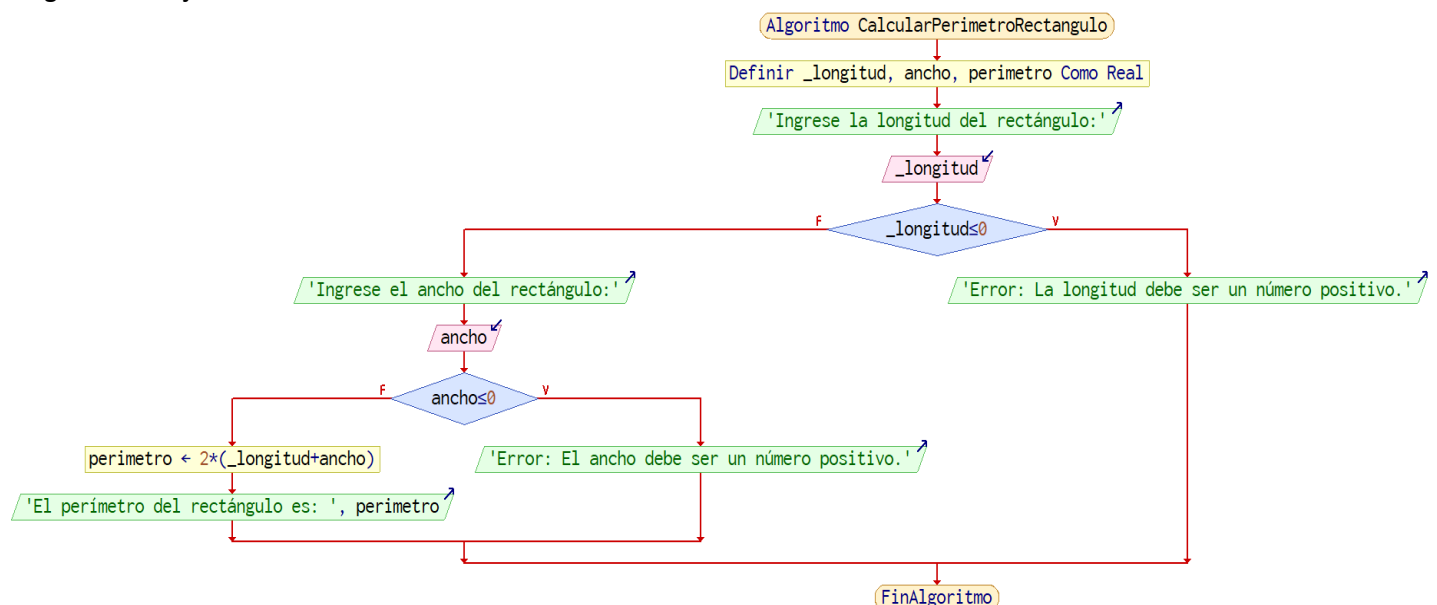
### Implementacion:

```

1  Algoritmo CalcularPerimetroRectangulo
2      // Declarar variables
3      Definir _longitud, ancho, perimetro Como Real
4      // Solicitar al usuario ingresar la longitud del rectángulo
5      Escribir 'Ingrese la longitud del rectángulo:'
6      Leer _longitud
7      // Verificar si la longitud es un número positivo
8      Si _longitud ≤ 0 Entonces
9          Escribir 'Error: La longitud debe ser un número positivo.'
10     SiNo
11         // Solicitar al usuario ingresar el ancho del rectángulo
12         Escribir 'Ingrese el ancho del rectángulo:'
13         Leer ancho
14         // Verificar si el ancho es un número positivo
15         Si ancho ≤ 0 Entonces
16             Escribir 'Error: El ancho debe ser un número positivo.'
17         SiNo
18             // Calcular el perímetro del rectángulo
19             perimetro ← 2*(_longitud+ancho)
20             // Mostrar el resultado
21             Escribir 'El perímetro del rectángulo es: ', perimetro
22         FinSi
23     FinSi
24 FinAlgoritmo
25

```

### Diagrama de flujo:



**Problema: Calcular del factorial**

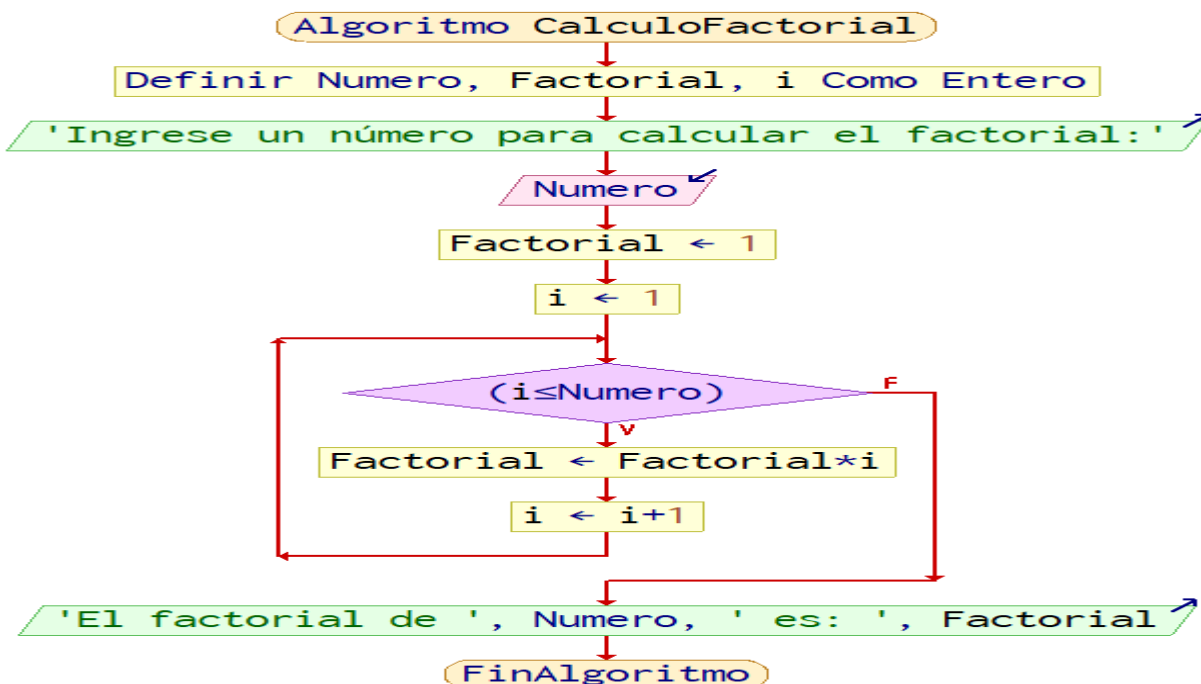
Diseña un programa que solicite al usuario ingresar un número entero no negativo y calcule su factorial. Asegúrate de manejar correctamente los casos en que el usuario ingrese un número negativo. Si el usuario proporciona un número negativo, muestra un mensaje de error indicando que el factorial no está definido para números negativos. Si el usuario ingresa un número no negativo, calcula el factorial utilizando un bucle mientras y muestra el resultado.

**Implementacion:**

```

1  Algoritmo CalculoFactorial
2      Definir Numero, Factorial, i Como Entero
3
4      Escribir "Ingrese un número para calcular el factorial:"
5      Leer Numero
6
7      // Inicializar la variable Factorial
8      Factorial = 1
9
10     // Bucle Mientras para calcular el factorial
11     i = 1
12     Mientras (i ≤ Numero) Hacer
13         Factorial = Factorial * i
14         i = i + 1
15     FinMientras
16
17     Escribir "El factorial de ", Numero, " es: ", Factorial
18
19 FinAlgoritmo
20

```

**Digrama de flujo:**

### Problema: Simulacion de lanzamiento de proyectil

Diseña un programa que simule el lanzamiento de un proyectil. El usuario debe ingresar la velocidad inicial del proyectil en metros por segundo y el ángulo de lanzamiento en grados. El programa calculará y mostrará la posición del proyectil en intervalos de tiempo, incluyendo la altura y la distancia recorrida. La simulación deberá detenerse y mostrar un mensaje cuando el proyectil alcance el suelo o alcance su altura máxima.

Asegúrate de manejar correctamente las unidades y de informar al usuario sobre la finalización de la simulación, ya sea porque el proyectil alcanzó el suelo o su altura máxima.

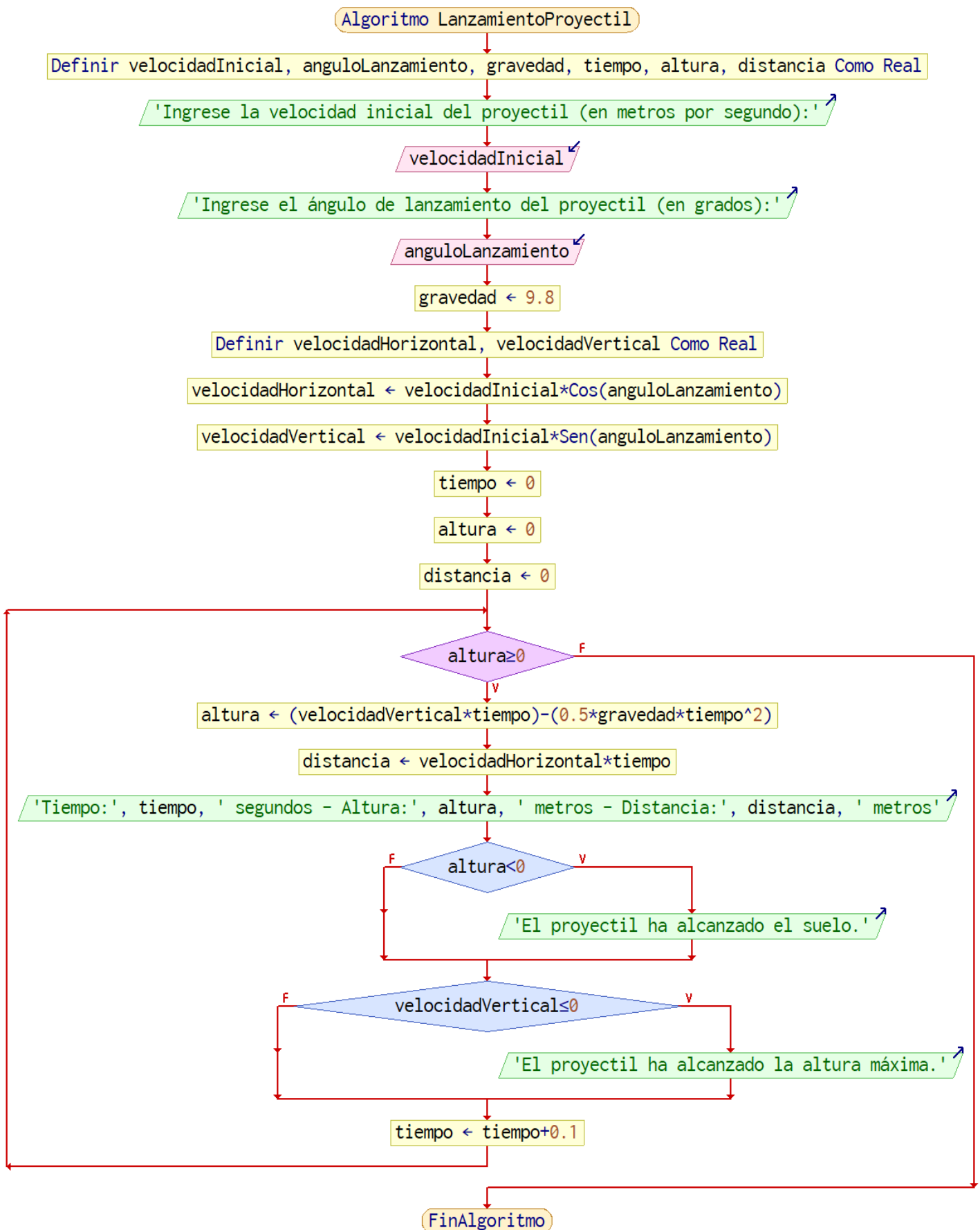
#### Implementacion:

```

1  Algoritmo LanzamientoProyectil
2      // Declarar variables
3      Definir velocidadInicial, anguloLanzamiento, gravedad, tiempo, altura, distancia Como Real
4      // Solicitar al usuario ingresar la velocidad inicial del proyectil
5      Escribir 'Ingrese la velocidad inicial del proyectil (en metros por segundo):'
6      Leer velocidadInicial
7      // Solicitar al usuario ingresar el ángulo de lanzamiento
8      Escribir 'Ingrese el ángulo de lanzamiento del proyectil (en grados):'
9      Leer anguloLanzamiento
10     // Inicializar constantes
11     gravedad ← 9.8
12     // Calcular componentes horizontal y vertical de la velocidad inicial
13     Definir velocidadHorizontal, velocidadVertical Como Real // Aceleración debido a la gravedad en la Tierra
14     velocidadHorizontal ← velocidadInicial*Cos(anguloLanzamiento)
15     velocidadVertical ← velocidadInicial*Sen(anguloLanzamiento)
16     // Inicializar variables de tiempo, altura y distancia
17     tiempo ← 0
18     altura ← 0
19     distancia ← 0
20     // Simular el lanzamiento del proyectil en intervalos de tiempo
21     Mientras altura ≥ 0 Hacer
22         // Calcular nueva posición del proyectil en el tiempo actual
23         altura ← (velocidadVertical*tiempo)-(0.5*gravedad*tiempo²)
24         distancia ← velocidadHorizontal*tiempo
25         // Mostrar posición actual del proyectil
26         Escribir 'Tiempo:', tiempo, ' segundos - Altura:', altura, ' metros - Distancia:', distancia, ' metros'
27         // Verificar si el proyectil ha alcanzado el suelo
28         Si altura < 0 Entonces
29             Escribir 'El proyectil ha alcanzado el suelo.'
30         FinSi
31         // Verificar si el proyectil ha alcanzado una altura máxima
32         Si velocidadVertical ≤ 0 Entonces
33             Escribir 'El proyectil ha alcanzado la altura máxima.'
34         FinSi
35         // Incrementar el tiempo en intervalos pequeños
36         tiempo ← tiempo+0.1
37     FinMientras
38 FinAlgoritmo

```

Digrama de flujo:



## Problema: Calcular el área de un triángulo

Diseña un programa que solicite al usuario ingresar la longitud de la base y la altura de un triángulo. El programa debe validar que ambos valores sean números positivos. Si el usuario ingresa valores no válidos, muestra un mensaje de error indicando que se deben ingresar valores numéricos válidos para la base y la altura. Si los valores son válidos, calcula el área del triángulo utilizando la fórmula  $\text{Area} = (\text{base} \times \text{altura}) / 2$  y muestra el resultado.

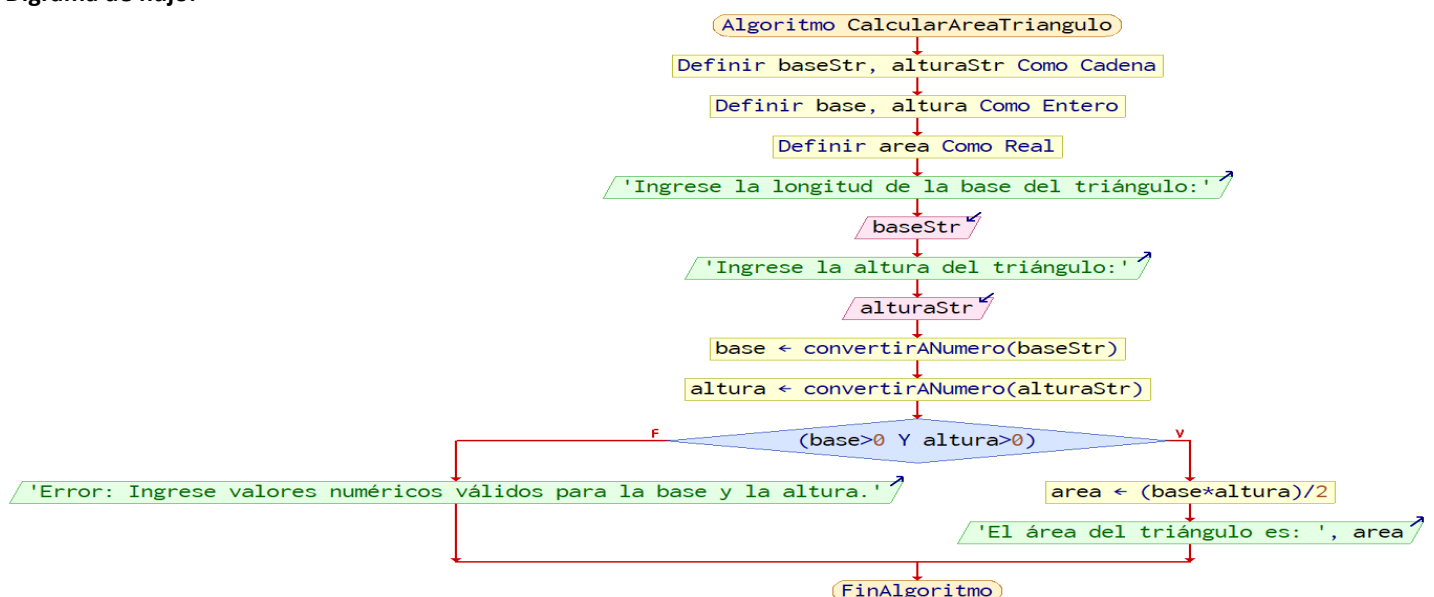
### Implementacion:

```

1  Algoritmo CalcularAreaTriangulo
2      // Declarar variables
3      Definir baseStr, alturaStr Como Cadena
4      Definir base, altura Como Entero
5      Definir area Como Real
6      // Solicitar al usuario que ingrese la base del triángulo
7      Escribir 'Ingrese la longitud de la base del triángulo:'
8      Leer baseStr
9      // Solicitar al usuario que ingrese la altura del triángulo
10     Escribir 'Ingrese la altura del triángulo:'
11     Leer alturaStr
12     // Convertir las cadenas ingresadas a enteros
13     base ← convertirANumero(baseStr)
14     altura ← convertirANumero(alturaStr)
15     // Verificar si base y altura son números mayores a 0
16     Si (base>0 Y altura>0) Entonces
17         // Calcular el área del triángulo
18         area ← (base*altura)/2
19         // Mostrar el resultado
20         Escribir 'El área del triángulo es: ', area
21     SiNo
22         // Mostrar mensaje de error si algún valor no es un número válido
23         Escribir 'Error: Ingrese valores numéricos válidos para la base y la altura.'
24     FinSi
25 FinAlgoritmo

```

### Digrama de flujo:



## Problema: Planificación de actividades

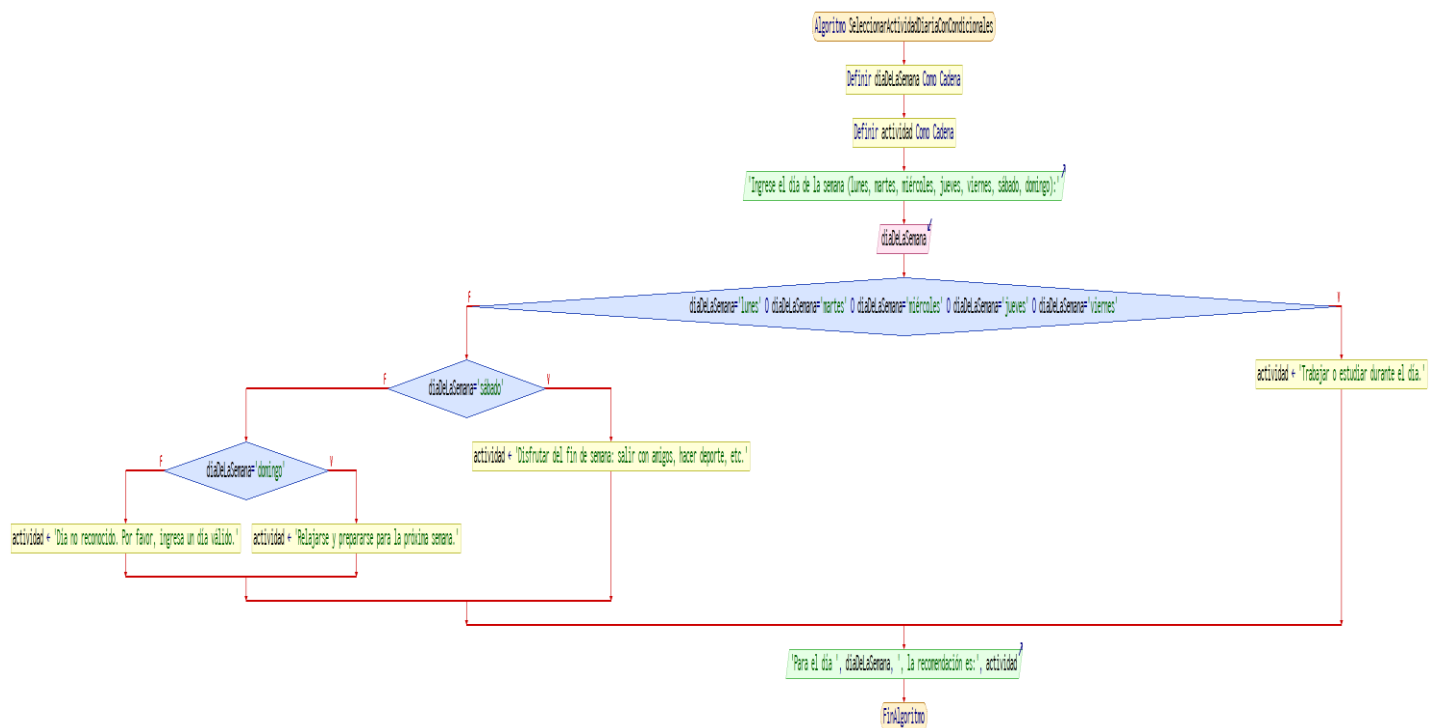
Diseña un programa que ayude a una persona a planificar sus actividades diarias para la semana. El usuario debe ingresar el día de la semana, y el programa le proporcionará una recomendación sobre qué actividad realizar con base en el algoritmo implementado.

### Implementación:

```

Algoritmo SeleccionarActividadDiariaConCondicionales
  // Declarar variables
  Definir diaDeLaSemana Como Cadena
  Definir actividad Como Cadena
  // Solicitar al usuario ingresar el día de la semana
  Escribir 'Ingrese el día de la semana (lunes, martes, miércoles, jueves, viernes, sábado, domingo):'
  Leer diaDeLaSemana
  // Utilizar condicionales para seleccionar una actividad
  Si diaDeLaSemana='lunes' O diaDeLaSemana='martes' O diaDeLaSemana='miércoles' O diaDeLaSemana='jueves' O diaDeLaSemana='viernes' Entonces
    actividad ← 'Trabajar o estudiar durante el día.'
  SiNo
    Si diaDeLaSemana='sábado' Entonces
      actividad ← 'Disfrutar del fin de semana: salir con amigos, hacer deporte, etc.'
    SiNo
      Si diaDeLaSemana='domingo' Entonces
        actividad ← 'Relajarse y prepararse para la próxima semana.'
      SiNo
        actividad ← 'Día no reconocido. Por favor, ingresa un día válido.'
      FinSi
    FinSi
  FinSi
  // Mostrar la actividad seleccionada
  Escribir 'Para el día ', diaDeLaSemana, ', la recomendación es:', actividad
FinAlgoritmo
  
```

### Diagrama de flujo:



## Problema: Cajero de tienda

Una tienda ha implementado un cajero automático para agilizar el proceso de pago de sus clientes. El cajero debe calcular el vuelto a entregar al cliente utilizando las denominaciones de billetes disponibles (100, 50, 25, 5, 1 dólares). Escribe un algoritmo que permita al cajero automatizado realizar el cálculo del vuelto de la siguiente manera:

- Solicita al usuario ingresar el precio total de los artículos comprados.
- Pide al usuario ingresar la cantidad entregada por el cliente.
- Verifica que el precio del artículo no sea mayor que la cantidad entregada. En caso de que lo sea, muestra un mensaje de error y finaliza el programa.
- Si el precio es menor o igual a la cantidad entregada, calcula y muestra al cliente el vuelto utilizando el menor número posible de billetes en las denominaciones mencionadas.

## Implementacion:

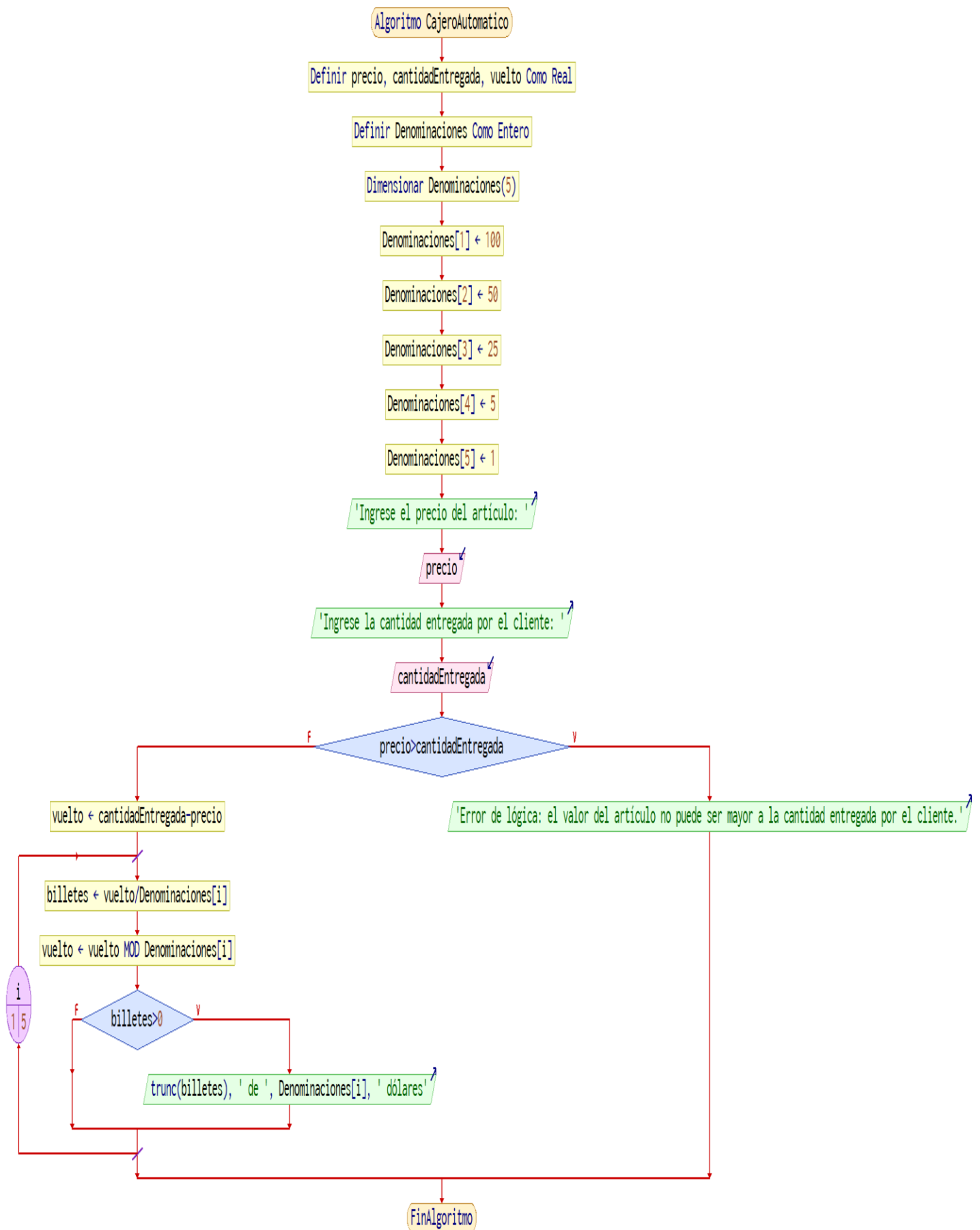
```

1  Algoritmo CajeroAutomatico
2      Definir precio, cantidadEntregada, vuelto Como Real
3
4      // Definir las denominaciones de billetes en un arreglo
5      Definir Denominaciones Como Entero
6      Dimension Denominaciones[5]
7      Denominaciones[1]=100
8      Denominaciones[2]=50
9      Denominaciones[3]=25
10     Denominaciones[4]=5
11     Denominaciones[5]=1
12
13     Escribir "Ingrese el precio del artículo: "
14     Leer precio
15
16     Escribir "Ingrese la cantidad entregada por el cliente: "
17     Leer cantidadEntregada
18
19     // Verificar si el precio del artículo es mayor que la cantidad entregada
20     Si precio > cantidadEntregada Entonces
21         Escribir "Error de lógica: el valor del artículo no puede ser mayor a la cantidad entregada por el cliente."
22
23     // Continuar con el cálculo del vuelto solo si la verificación es correcta
24     Sino
25         vuelto <- cantidadEntregada - precio
26         // Bucle para calcular la cantidad de cada tipo de billete
27         Para i <- 1 Hasta 5 Hacer
28             // Calcular la cantidad de billetes de la denominación actual
29             billetes <- vuelto / Denominaciones[i]
30             vuelto <- vuelto % Denominaciones[i]
31
32             // Mostrar la cantidad de billetes si es mayor que cero
33             Si billetes > 0 Entonces
34                 Escribir trunc(billetes), " de ", Denominaciones[i], " dólares"
35             FinSi
36         FinPara
37     FinSi
38 FinAlgoritmo

```



Digrama de flujo:



**Problema: Pago de salarios**

Una empresa paga a sus 100 operarios semanalmente, de acuerdo con el número de horas trabajadas, a razón de P pesetas por hora y de 1.5 pesetas por cada hora extra. Las horas extras son las que exceden de 40h. Hacer el organigrama que dado el número de horas trabajadas y el valor de P calcule el salario que le corresponde.

**Implementacion:**

```

1  Algoritmo CalcularSalario
2
3  // Declaración de variables
4  Definir horas_trabajadas, valor_pesetas, salario_base, horas_extra, salario_total Como Entero
5
6  // Solicitar al usuario el número de horas trabajadas
7  Escribir "Ingrese el número de horas trabajadas:"
8  Leer horas_trabajadas
9
10 // Solicitar al usuario el valor de Pesetas
11 Escribir "Ingrese el valor de Pesetas por hora:"
12
13
14 // Verificar si las horas trabajadas son menores o iguales a 40
15 Si horas_trabajadas ≤ 40 Entonces
16     // Calcular el salario base sin horas extra
17     salario_base = horas_trabajadas * valor_pesetas
18     // No hay horas extra trabajadas
19     horas_extra = 0
20 Sino
21     // Calcular el salario base considerando 40 horas
22     salario_base = 40 * valor_pesetas
23     // Calcular las horas extra trabajadas
24     horas_extra = horas_trabajadas - 40
25 FinSi
26
27 // Calcular el salario total considerando las horas extra
28 salario_total = salario_base + (horas_extra * 1.5 * valor_pesetas)
29
30 // Mostrar el salario total
31 Escribir "El salario correspondiente es:", salario_total
32
33 FinAlgoritmo
34

```

**Digrama de****flujo:**

