

## 2. ALGORITMOS Y PSEUDOCODIGOS

### 2.1 CARACTERÍSTICAS DE UN ALGORITMO

Un Algoritmo es una secuencia de pasos para resolver un problema y debe contar con las siguientes características:

**Preciso:** cada paso debe ser claro y exacto en su construcción para que así determine puntualmente lo que se desea hacer.

**Definido:** toda vez que se ejecute el algoritmo con los mismos datos de entrada, éste debe generar el mismo resultado.

**Finito:** todo algoritmo debe tener un fin.

El algoritmo se construye usando palabras del idioma y debe poder ser entendido por cualquier persona. Cuando se utilizan líneas de código se crea un programa y éstos se escriben usando lenguajes de programación. Si se emplea cuasi código en la construcción del programa, se tiene un Pseudocódigo.

Un algoritmo se construye en forma general usando los siguientes pasos:

1. Inicio.
2. Capturar, conocer o ingresar todos los datos que permitan resolver el problema.
3. Resolver, calcular o llevar a cabo los procesos con los datos capturados. Se sugiere en este punto colocar las ecuaciones necesarias.
4. Mostrar, visualizar o imprimir todos los resultados que el problema exija.
5. Fin.

El siguiente ejemplo sencillo muestra un error clásico en la elaboración de un algoritmo:

Construya un algoritmo que encuentre y muestre el valor del área de un rectángulo.

**PASOS**

- 1 Conocer los valores de altura y base
- 2 Calcular el área
- 3 Mostrar el valor del área hallado

El algoritmo no es preciso en sus pasos porque no indica a que figura geométrica hay que calcularle el área, por lo tanto no se puede identificar la fórmula a aplicar. Además, para mostrar que un algoritmo termina es conveniente incluir como último paso la palabra fin.

El algoritmo para resolver el ejemplo anterior sería entonces:

**PASOS**

- 1 Inicio
- 2 Conocer los valores de altura y base del rectángulo
- 3 Calcular el área del rectángulo
- 4 Mostrar el valor del área del rectángulo
- 5 Fin

## 2.2 TIPOS DE INSTRUCCIONES

A partir de los algoritmos se pueden elaborar programas y para ello simplemente basta con “convertir” los pasos del algoritmo a sus instrucciones respectivas según el lenguaje de programación que se esté utilizando pero, como existen diferentes lenguajes, inicialmente se construirán los programas usando una de las herramientas de programación: el Pseudocódigo.



El Pseudocódigo se construye con instrucciones que no son específicas de un lenguaje de programación pero que reflejan un comportamiento similar; al tener un programa en pseudocódigo se puede convertir éste con mayor facilidad a un lenguaje de programación.

Las siguientes son algunas de las instrucciones que hacen parte del pseudocódigo.

### 2.2.1 Instrucción Inicio / Fin

Se utilizan para dar comienzo y terminación al pseudocódigo, por cada programa escrito debe existir un solo **Inicio** y **Fin**. A continuación se muestran como se escriben las instrucciones en el pseudocódigo y a su vez, su representación en la segunda herramienta de programación: el diagrama de flujo, ver Tabla 7.

*Tabla 7. Representación de las instrucciones Inicio y Fin en el diagrama de flujo.*

PSEUDOCÓDIGO Instrucción	DIAGRAMA DE FLUJO
Inicio	
Fin	

Fuente: Autores

**Inicio** y **Fin** son Palabras Reservadas, es decir, palabras que pueden ser usadas únicamente para cumplir una función específica, por tanto, éstas no pueden ser usadas como identificadores.

### 2.2.2 Instrucción de Asignación

Esta instrucción permite representar las operaciones aritméticas en el mundo del pseudocódigo.

Siempre que se desee realizar un procedimiento, operación o cálculo – con operadores aritméticos, relacionales y/o lógicos –, se debe utilizar una instrucción de Asignación.

En el mundo de las matemáticas se puede encontrar una expresión como la siguiente:

$$Y = X + 2$$

En donde para saber el valor de Y primero se debe conocer a X y luego llevar a cabo la operación de suma.

En el mundo del pseudocódigo la anterior expresión se convierte a:

$Y \leftarrow X + 2$



LADO DERECHO

El símbolo Flecha representa la instrucción de Asignación.

Para conocer el valor de Y primero se debe resolver el lado derecho de la instrucción. En forma general se interpreta toda instrucción de Asignación de la siguiente manera:

**“El resultado del lado derecho de la flecha se almacena, guarda o asigna en el lado izquierdo”.**

1. Las letras al lado **izquierdo** de la asignación, como en el ejemplo anterior la **Y**, representan un identificador.
2. Las letras al lado **derecho** de la asignación puede representar dos cosas:
  - a. Si no llevan comillas, como en el ejemplo anterior la **X**, son un identificador.
  - b. Si llevan comillas (simples o dobles), son un dato tipo carácter. En el ejemplo:  $Y \leftarrow 'X'$ , la letra **X** se almacena en **Y**.

Para poder emplear identificador dentro de una expresión, éste debe:

1. Estar declarado: lo que significa que el computador sabe que existe un espacio de memoria con ese nombre y conoce el tipo de dato que se almacenará en él.
2. Estar inicializado: es necesario que los identificadores del lado derecho tengan un valor inicial para emplearlo en la expresión. Por lo tanto la secuencia de instrucciones correcta sería:

$X \leftarrow 3$   
 $Y \leftarrow X + 2$

3. Corresponder en tipo de dato: tanto los identificadores de la derecha como los de la izquierda deben almacenar datos del mismo tipo.

Ejemplos:

REPRESENTACIÓN DE LA EJECUCIÓN		
INSTRUCCIÓN	Ejecución de las instrucciones	Id
	X y Y son de tipo entero	
X ← 3	Al ejecutarse se almacena en X el valor de 3, así:	X 3
Y ← X + 2	El contenido de la celda X se suma con el número 2 y se almacena en Y, entonces en Y se guarda el resultado, el número 5, así:	Y 5

REPRESENTACIÓN DE LA EJECUCIÓN		
INSTRUCCIÓN	Ejecución de las instrucciones	Id
	X y Y son de tipo decimal	
X ← 3	Al ejecutarse se almacena en X el valor de 3, así:	X 3
Y ← X + 2	El contenido de la celda X se suma con el número 2 y se almacena en Y, entonces en Y se guarda el resultado, el número 5, así:	Y 5
Y ← 2.89	Al ejecutarse se almacena en Y el valor de 2.89 esto ocasiona que el anterior valor de Y (5) se pierda y sea reemplazado por uno nuevo (2.89)	Y 2.89
	Note en este ejemplo que se manejan datos enteros y decimales en la misma variable. En la ejecución de las instrucciones no se genera error ya que los números decimales contienen a los números enteros.	

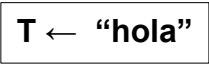
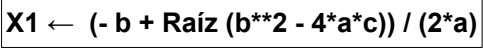

REPRESENTACIÓN DE LA EJECUCIÓN				
INSTRUCCIÓN	Ejecución de las instrucciones	Id		
	X es de tipo entero			
$X \leftarrow 3$	Al ejecutarse se almacena en X el valor de 3, así:	<table border="1"> <tr><td>X</td></tr> <tr><td>3</td></tr> </table>	X	3
X				
3				
$X \leftarrow "2"$	Al ejecutarse se genera un error, ya que el tipo de dato ahora es carácter y la variable sólo recibe números enteros			

REPRESENTACIÓN DE LA EJECUCIÓN				
INSTRUCCIÓN	Ejecución de las instrucciones	Id		
	T es de tipo cadena			
$T \leftarrow "hola"$	Al ejecutarse se almacena en T el dato "hola". Note como el identificador T está conformada por 4 celdas para poder almacenar la palabra	<table border="1"> <tr><td>T</td></tr> <tr><td>h o l a</td></tr> </table>	T	h o l a
T				
h o l a				
$T \leftarrow "ola"$	El contenido de la celda T se destruye y se guarda la palabra "ola"	<table border="1"> <tr><td>T</td></tr> <tr><td>o l a</td></tr> </table>	T	o l a
T				
o l a				
$T \leftarrow "holas"$	No se puede ejecutar ya que el dato "holas" está conformado por 5 letras y T representa sólo 4 celdas			

REPRESENTACIÓN DE LA EJECUCIÓN				
INSTRUCCIÓN	Ejecución de las instrucciones	Id		
$X \leftarrow 7$	X es de tipo entero. Al ejecutarse se almacena en X el valor de 7, así:	<table border="1"> <tr><td>X</td></tr> <tr><td>7</td></tr> </table>	X	7
X				
7				
$Y \leftarrow '8'$	Y es de tipo carácter. Al ejecutarse se almacena en Y el valor de '8', así:	<table border="1"> <tr><td>Y</td></tr> <tr><td>8</td></tr> </table>	Y	8
Y				
8				
$Z \leftarrow X + Y$	Esta instrucción no se puede ejecutar en el mundo del pseudocódigo, porque no se pueden sumar enteros y caracteres			

La instrucción de Asignación utiliza la figura del rectángulo para su representación en el diagrama de flujo, en la Tabla 7 se muestran algunos ejemplos:

**Tabla 7. Representación de la instrucción de Asignación en el diagrama de flujo.**

PSEUDOCÓDIGO Instrucción	DIAGRAMA DE FLUJO
$T \leftarrow \text{"hola"}$	
$X1 \leftarrow (-b + \text{Raíz}(b^{**}2 - 4*a*c)) / (2*a)$	
$\text{Suma} \leftarrow \text{Suma} + \text{nota}$	

Fuente: Autores

### 2.2.3 Instrucción de Lectura

La gran mayoría de problemas necesitan para ser resueltos, datos ingresados por el usuario y éstos deben ser capturados por el programa, en el mundo del pseudocódigo existe una instrucción que permite hacer esto, la instrucción **Leer**. Entonces **Leer** es una palabra reservada.

La instrucción **Leer** tiene el siguiente formato general en el pseudocódigo

$Y \leftarrow X + 2$



**LADO DERECHO**

Dentro del paréntesis se debe colocar el identificador en donde se guardará la información a capturar. Para poder ejecutar una instrucción **Leer** el usuario debe ingresar el dato, el computador lo captura y automáticamente lo almacena en la celda correspondiente.

La instrucción **Leer** puede construirse de dos maneras:

#### 1. Leer ( A )

Cuando se ejecuta esta instrucción, en **A** se almacena el dato ingresado por el usuario.

## 2. Leer ( a, b, c)

Si se desea capturar más de un dato a la vez, se debe colocar dentro de los paréntesis, separados por comas, una cantidad de identificadores igual a la cantidad de datos a capturar.

En el ejemplo, **a**, **b** y **c** son tres identificadores, por tanto, se necesitan ingresar tres datos.

Toda instrucción **Leer** debe cumplir con lo siguiente:

1. Lo(s) identificador(es) que se encuentre(n) dentro de los paréntesis, debe(n) estar declaradas.
2. Debe existir coincidencia en el tipo de dato que se va a capturar con el tipo de dato que puede almacenar la celda que se ha colocado dentro de los paréntesis.
3. Cuando se tiene más de una celda dentro de los paréntesis de la instrucción **Leer**, se debe tener mucho cuidado en cómo se le sugiere al usuario el ingreso de los datos para que cuando los ingrese, éstos coincidan con la cantidad de celdas y el tipo de dato que puede guardar cada una de ellas.

Por ejemplo: si se quiere capturar los datos de peso, edad y altura en cm de una persona se construye la instrucción **Leer** de la siguiente manera – para el ejemplo se asume que todas las celdas son de tipo entero –:

**Leer** (peso, edad, altura)

Si una persona tiene:           Peso 60 kilos  
  Edad 26 años  
  Altura 150 cm

Pero los digita de la siguiente manera:

60 150 26

Al ejecutarse la instrucción **Leer**, automáticamente se guarda 60 en peso, 150 en edad y, 26 en altura. Al computador no le preocupa que la persona tenga 60 kilos de peso, 150 años y 26 cm. de altura, esos fueron los datos digitados y con ellos el programa trabajará. El computador no generará error alguno sin embargo los resultados obtenidos cuando se ejecute el programa serán incorrectos.



Para ser más claro, una alternativa a la instrucción **Leer** (peso, edad, altura) sería:

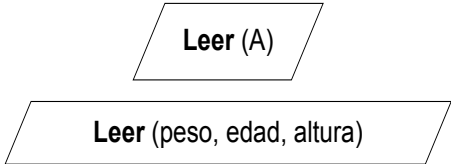
**Leer** ( peso )

**Leer** ( edad )

**Leer** ( altura )

La instrucción **Leer** se representa con un romboide en el mundo del diagrama de flujo, a continuación, ver la Tabla 8 con algunos ejemplos:

**Tabla 8. Representación de la instrucción Leer en el diagrama de flujo.**

PSEUDOCÓDIGO Instrucción	DIAGRAMA DE FLUJO
<p style="text-align: center;"><b>Leer</b> (A)</p> <p style="text-align: center;"><b>Leer</b> (peso, edad, altura)</p>	

Fuente: Autores

La instrucción **Leer** se necesita en casi todos los problemas, a continuación se muestran algunos ejemplos de enunciados típicos y sus características:

- Aquellos en los cuales aparece la palabra conocidos, dados, ingresar o capturar:

Dados tres números construya un pseudocódigo que muestre el mayor de ellos.

Para poder resolver el problema es necesario conocer los valores de los tres números y éstos deben ser suministrados por el usuario.

Escriba un Programa que lea un número entero N y calcule el resultado

de la siguiente serie:  $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{N}$

En este segundo ejemplo, se conocen como deben ser los términos a generar sin embargo, la cantidad de los mismos está determinada por el valor que tome N, el cual debe ser ingresado por el usuario. Si el usuario ingresa el número 3, el programa debe mostrar el resultado de sumar

$1 - \frac{1}{2} + \frac{1}{3}$  es decir, 0.8333333.

- Problemas que resuelven una ecuación cuya solución se obtiene por reemplazar en ella los identificadores por números.

Construya un pseudocódigo que muestre el resultado de las raíces reales de la siguiente expresión  $aX^2 + bX + c = 0$

Para encontrar la solución de las raíces reales se necesita conocer los valores de a, b y c.

1. Aquellos problemas cuya solución sólo se puede obtener por el proceso que se le haga a los datos suministrados por el usuario. Por ejemplo:

Diseñe un algoritmo que determine si un número es o no es, par positivo.

Para resolver el problema, es necesario que el usuario ingrese el número a evaluar y así determinar si éste es o no un par positivo.

Normalmente la gran mayoría de los enunciados muestran la necesidad de solicitar datos al usuario.

#### 2.2.4 Instrucción de Escritura

Durante la solución de un problema a través de pseudocódigo, generalmente se necesita mostrar resultados parciales o totales, esto se logra con la instrucción **Escribir**. En forma general siempre que se desee mostrar, visualizar o imprimir los resultados, en el mundo del pseudocódigo se usa la instrucción **Escribir**.

La instrucción **Escribir** tiene el siguiente formato:

**Escribir** (     )  
**Aquí se puede colocar  
identificadores, texto o ambos**

Dentro de los paréntesis se pueden colocar cadenas de texto, identificadores o combinación de ambos. **Escribir** es una palabra reservada. Toda instrucción **Escribir** al ser ejecutada visualizará en pantalla lo que se ha colocado dentro de los paréntesis.

La instrucción **Escribir** tiene los siguientes formatos:

### 1. Escribir ( A )

Cuando se ejecuta esta instrucción el computador muestra el contenido del identificador siempre y cuando esté declarado y contenga un dato.

### 2. Escribir ( “ texto “ )

Siempre que se desee mostrar algún comentario, información o texto en general el uso de este tipo de formato es el adecuado. Este tipo de formato no requiere cumplir con condición alguna.

### 3. Escribir ( “ texto ”, A )

Si se desea acompañar el contenido de una celda con un texto, se emplea este formato. El identificador debe estar declarado y contener un dato.

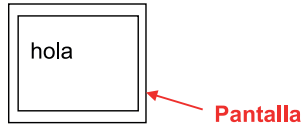
En la Tabla 9 se muestran algunos ejemplos, en todos ellos se asume que los identificadores ya están declaradas.

**Tabla 9. Ejemplos de Instrucciones y ejecución**

INSTRUCCIÓN	REPRESENTACIÓN DE LA EJECUCIÓN						
	Ejecución de las instrucciones	Celda					
<b>T</b> ← “hola” <b>Escribir</b> ( T )	Al ejecutarse se almacena en T el dato cadena “hola”, Cuando se ejecuta la instrucción <b>Escribir</b> aparece en pantalla lo siguiente:	<div style="text-align: center;">                         T  <table border="1" style="margin: auto;"> <tr> <td>h</td> <td>o</td> <td>l</td> <td>a</td> </tr> </table>   <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">hola</td> </tr> </table>   <span style="color: red;">Pantalla</span> →                     </div>	h	o	l	a	hola
h	o	l	a				
hola							
<b>G</b> ← 535 <b>F</b> ← “es” <b>Escribir</b> ( F, G )	En la <b>G</b> se almacena el valor 535 En la <b>F</b> se almacena el dato es Cuando se ejecuta la instrucción <b>Escribir</b> se muestra en pantalla:	<div style="text-align: center;">                         G  <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">535</td> </tr> </table>   <table border="1" style="margin: auto;"> <tr> <td>e</td> <td>s</td> </tr> </table>   <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">es 535</td> </tr> </table>   <span style="color: red;">Pantalla</span> →                     </div>	535	e	s	es 535	
535							
e	s						
es 535							

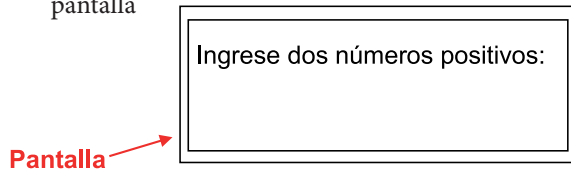
**Escribir** (“hola”)

Cuando se ejecuta la instrucción **Escribir** aparece en pantalla lo siguiente:



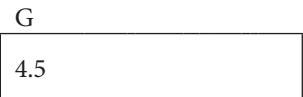
**Escribir** (“Ingrese dos números positivos:”)

Cuando se ejecuta la instrucción **Escribir** se muestra en pantalla



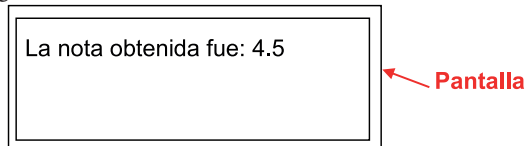
$G \leftarrow 4.5$

Se almacena 4.5 en G



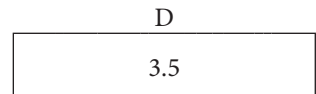
**Escribir** (“La nota obtenida fue:”, G)

Cuando se ejecuta la instrucción **Escribir** aparece en pantalla lo siguiente:



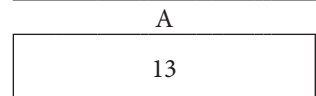
$D \leftarrow 3.5$

Cuando se ejecuta se almacena en D 3.5



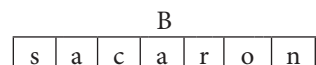
$A \leftarrow 13$

Cuando se ejecuta se almacena en A 13



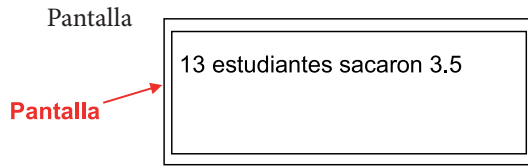
$B \leftarrow$  “sacaron”

Cuando se ejecuta se almacena en B sacaron



**Escribir** ( A , “estudiantes” , B, D )

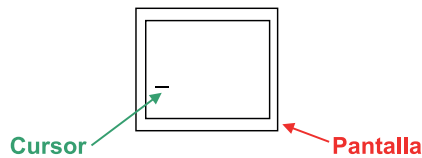
Cuando se ejecuta la instrucción **Escribir** se muestra en



Generalmente antes de una instrucción **Leer** va una instrucción **Escribir** esto con el fin de informar al usuario lo que se desea que éste ingrese. En el ejemplo del peso, edad y altura se creó la siguiente instrucción

**Leer** (peso, edad, altura)

Cuando el computador llega a esta instrucción en pantalla aparece lo siguiente:

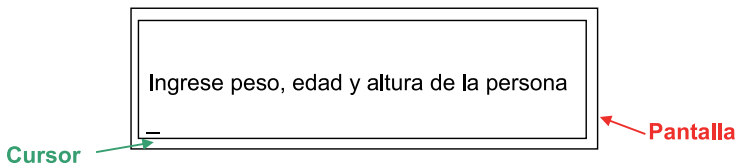


Justo antes de ejecutarse la instrucción **Leer**, en pantalla aparece el cursor titilando en espera a que usuario ingrese los datos. La pantalla con tan sólo un cursor titilando no ofrece información alguna al usuario acerca de qué debe hacer, es recomendado utilizar la instrucción **Escribir** para indicarle que datos suministrar. Agregando una instrucción **Escribir** antes de la instrucción **Leer** quedaría:

**Escribir** (“Ingrese peso, edad y altura de la persona”)

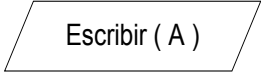

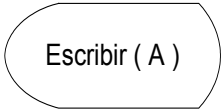
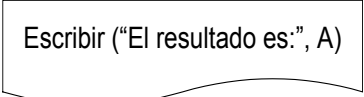
**Leer** (peso, edad, altura)

De esta manera en pantalla aparece:



En la Tabla 10 se puede apreciar tres tipos de símbolos para la instrucción **Escribir**, el primero de ellos es idéntico al usado por la instrucción **Leer**, el segundo, es usado para imprimir y, el tercero, para mostrar en pantalla. El símbolo depende del programa que emplee para generar el diagrama de flujo.

**Tabla 10. Representación de la instrucción Escribir en el diagrama de flujo.**

PSEUDOCODIGO Instrucción	DIAGRAMA DE FLUJO
<p style="text-align: center;"><b>Escribir (A)</b></p> <p><b>Escribir</b> (“El resultado es:”, A)</p>	<p style="text-align: center;">1. </p> <p style="text-align: center;">2. </p> <p style="text-align: center;">3. </p> <p style="text-align: center;"></p>

### 2.3 FORMATO GENERAL DEL PSEUDOCÓDIGO

Todo programa escrito en pseudocódigo está conformado por los siguientes tres bloques:

1. La Cabecera donde se coloca el nombre del programa escrito en pseudocódigo:

*Algoritmo\_nombre\_del\_programa*

2. El Bloque de Declaración es donde se informa al computador cuántas celdas y de qué tipo se van a necesitar, aquí se deben declarar todos los identificadores que se utilizarán en el transcurso de la ejecución del programa.

El Bloque de Declaración puede contener dos secciones: Declaración de Variables y Declaración de Constantes.

- a. *Declaración de variables*: comienza con la palabra reservada **var**, debajo de ella se debe colocar primero, el tipo de dato, seguido de dos puntos (:) y por último, la(s) celda(s) que se crearán.

```
var
      Tipo_de_dato : Listado_de_celdas
```

Si se desea declarar más de una variable del mismo tipo de dato, se deben separar con comas, por ejemplo:

```
var
      Entero : A, B
```

Lo anterior también se puede escribir como sigue:

```
var
      Entero : A
      Entero : B
```

El siguiente ejemplo muestra el uso de una celda para cada tipo de dato.

```
var
      Entero : A
      Decimal : B
      Caracter : C
      Booleano: G
```

Ningún nombre de variable se puede utilizar más de una vez. En el siguiente ejemplo la variable A se encuentra declarada dos veces, por lo tanto generará error en la ejecución del programa:

```
var
      Entero : A
      Caracter : A
      Booleano: G
```

Para declarar una variable cadena primero se le asigna un nombre y luego entre corchetes se coloca un número que representa la cantidad de celdas que usará la variable, por ejemplo:

**var**  
Cadena: D[10]

Las palabras **Decimal**, **Carácter**, **Entero**, **Cadena** y **Booleano** son palabras reservadas. El siguiente ejemplo muestra un error por usar una palabra reservada para una función a la cual no pertenece:

**var**  
Decimal : a, A, Entero

Existen dos maneras que ayudan en determinar cuántas variables y de qué tipos se necesitan declarar. La primera es la Técnica de las Preguntas que consiste en formularse dos preguntas cuyas respuestas ayudaran a construir la zona de Declaración de variables.

- *Pregúntese cuántos datos necesita conocer para resolver el problema.* La misma cantidad de datos obtenida en la respuesta corresponden a la cantidad de celdas que se deben crear para almacenarlos. A cada celda se le da un nombre y se evalúa qué tipo de dato debe contener.
- *Pregúntese cuántos cálculos necesita resolver con los datos capturados en la pregunta anterior.* La cantidad de cálculos a llevar a cabo es la misma cantidad de variables que se deben crear para almacenar los datos obtenidos al solucionar los mismos. A cada uno se le da un nombre y se evalúa de qué tipo de dato será el resultado obtenido.

La segunda técnica es la Técnica del Algoritmo, ésta consiste en guiarse por las palabras utilizadas en los pasos del algoritmo construido. En los ejercicios al final del capítulo se estudia en detalle cómo construir un algoritmo y, después, la aplicación de estas técnicas.

- b.** Declaración de constantes: comienza con la palabra reservada **const**, debajo de ella se debe colocar primero, el identificador, seguido del símbolo igual (=) y por último, el valor que se asignará a la celda. El valor es un dato, no una expresión aritmética que pueda generarlo.



**const**

← **Símbolo del IGUAL =**

**Identificador = Valor**

Nunca se puede utilizar un identificador, que se encuentre en la declaración de constantes, dentro de los paréntesis en una instrucción **Leer** ya que este espacio está reservado únicamente para las celdas colocadas en la zona de declaración de variables.

Si se quiere declarar la constante PI se debe hacer lo siguiente:

**const**

pi = 3.1416

Recuérdese que después del igual (=) debe ir el valor y no una expresión. No es válido lo siguiente en el mundo del pseudocódigo:

**const**

*raizdedos* =  $\sqrt{2}$

*cociente* =  $\frac{4}{5}$

*porcen1* = 0.05%

Algunos ejemplos de declaración de constantes serían:

**const**

pi = 3.1416

raizdedos = 1.4142

nombre = "pedro"

letra = 's'

porcent1 = 0.05

Cualquier variable declarada en la zona de declaración de variables **NO** puede ser utilizada en la de constantes. El siguiente ejemplo genera error al ser ejecutado puesto que A es al mismo tiempo una variable y una constante.

```
var
    Entero :A
    Decimal : B
const
    A = 3.456
    Suma = 45
```

La sección de declaración de constantes es opcional.

3. El Bloque de Ejecución es el cuerpo de instrucciones que debe ejecutar el computador. Comienza y termina con las palabras reservadas **Inicio** y **Fin**, y en su interior podrá contener la repetición de las otras instrucciones las veces que sea necesario.

Entonces el formato general del pseudocódigo es:

```
Algoritmo_nombre_del_programa } ← Cabecera
var
    Tipo_de_dato : lista de variables }
const
    Identificador = valor } ← Bloque de declaración

Inicio
    Instrucción 1
    :
    Instrucción N } ← Bloque de Ejecución
Fin
```

## 2.4 EJERCICIOS

### 2.4.1 Ejercicios con Respuesta

1. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que conocida la base y altura de un rectángulo, calcule y muestre su área y perímetro.
2. Realice un algoritmo, pseudocódigo y diagrama de flujo tal que, dado el nombre de una persona (de máximo 8 caracteres), el apellido (de máximo 6), la edad en años y su peso en libras, calcule y muestre la edad en número de días, el peso en kilogramos, y el apellido seguido del nombre.
3. La potencia que consume un bombillo se expresa por medio de la fórmula  $P = V \times I$ , donde  $V$  es el voltaje e  $I$  es la corriente. El voltaje está en función de la corriente  $I$  y la resistencia  $R$  del bombillo, por medio de la siguiente expresión:  $V = I \times R$ . La potencia total entregada por la fuente es igual a la suma de las potencias de todos los bombillos. La figura siguiente muestra cuatro (4) bombillos, y ellos están conectados a una fuente. Construya un algoritmo, pseudocódigo y diagrama de flujo tal que, conocidos el valor de la corriente  $I$  y los valores de resistencia  $R$  - de cada bombillo -, calcule y muestre la potencia de cada bombillo y la potencia total entregada por la fuente, ver Ilustración 1.

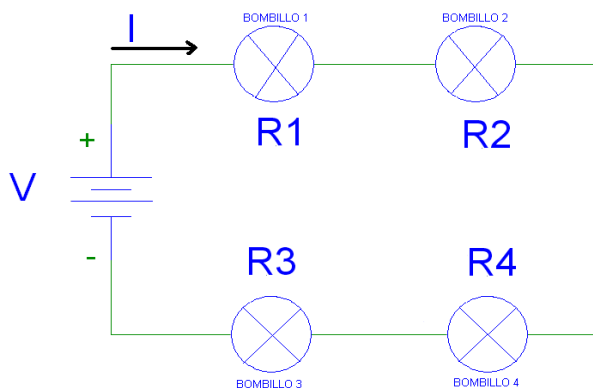


Ilustración 1. Circuito serie conformado por cuatro bombillos y una fuente de voltaje

4. En la tienda Wall Max existen los siguientes productos:

Artículo	Marca	Valor US\$
Camisa	<i>T-shirt</i>	\$15
Pantalón	<i>Lave`se</i>	\$25 el par
Zapatos	<i>Acme</i>	\$80
Ropa interior	<i>Proteggeme</i>	\$45
Medias	<i>Ropita pa pie</i>	\$10
DVD	<i>Pollito</i>	\$70

La tienda ofrece un descuento del: 10% en las camisas y 10% en los pantalones por la compra de 3 de cada uno de ellos; 4% por la compra de dos pares de zapatos y el 50% de descuento por el tercer par; por la compra de cinco pares de medias el descuento es del 5% y 8% de descuento por comprar dos DVD. Construya un algoritmo, pseudocódigo y diagrama de flujo que muestre: el valor total de cada artículo que se compra, el valor de cada descuento obtenido, el valor total a pagar.

#### 2.4.2 Ejercicios sin Respuesta

1. Pedro tiene el triple de la edad de Juan. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que dada la edad de Pedro, calcule y muestre la edad de Juan.
2. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que conocido el radio de un círculo, calcule y muestre su diámetro, circunferencia y área.
3. El supermercado Pepín se encuentra en promoción “miércoles de verduras y frutas”; descuento del 15% sobre las compras. Suponiendo que la suma de todas las verduras y frutas compradas fue **V**, construya un algoritmo, pseudocódigo y diagrama de flujo que calcule y muestre cuánto habría que pagar si hoy fuera miércoles.
4. Su profesor de programación ha sacado 4 notas a lo largo del semestre y es el momento de conocer su definitiva. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que dadas unas 4 notas, calcule y muestre el promedio de las mismas.
5. Se necesitan 20 ladrillos para cubrir un área de 1 m<sup>2</sup>; suponiendo que hay que cubrir un área de **X** metros \* **Y** metros; construya un algoritmo,

pseudocódigo y diagrama de flujo que calcule y muestre cuantos ladrillos se necesitarían en total para cubrir toda el área.

6. Usted saca un crédito en el banco a 5 años con un interés anual del 3%. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que conocido el monto inicial del préstamo, calcule y muestre cuanto debe pagar al cabo de los 5 años.

### 2.4.3 Respuesta a los Ejercicios

1. Construya un algoritmo, pseudocódigo y diagrama de flujo tal, que conocida la base y altura de un rectángulo, calcule y muestre su área y perímetro.
  - Lo primero es comprender el problema, para ello se conceptualiza y determina el objetivo:
  - Conceptualización: el problema consiste en obtener de una figura geométrica conocida - el rectángulo - su área y perímetro. Para ello se deben utilizar las ecuaciones:

$$A = B \times H \qquad P = 2(B + H)$$

Objetivo: calcular el valor del área y del perímetro de un rectángulo, utilizando la base y altura ingresadas por el usuario.

- Una vez comprendido el problema, se procede a construir el algoritmo:

Para resolver el problema se requiere que el usuario ingrese la base y la altura, por lo tanto deben ser capturados. Luego, se resuelven las ecuaciones del área y perímetro con los datos capturados. Y finalmente, se muestran los resultados. El algoritmo queda así:

1. Inicio
2. Capturar la base y altura del rectángulo
3. Resolver las ecuaciones del área y perímetro del rectángulo
$$A = B \times H \qquad P = 2(B + H)$$
4. Mostrar los resultados del área y perímetro del rectángulo
5. Fin

- Una vez construido el algoritmo el siguiente paso es elaborar el pseudocódigo:

Se determina el nombre que se va a dar al programa y se coloca en la cabecera:

Algoritmo\_Calculo\_Area\_y\_Perimetro\_Rectangulo

Para determinar lo que se debe colocar en la zona de declaración de variables se tendrá en cuenta la técnica de las preguntas:

*¿Cuántos datos necesita conocer para resolver el problema?* Resolver el problema implica calcular el área y perímetro del rectángulo y para lograr esto se necesita conocer dos valores: la **base** y la **altura**, entonces se necesita crear dos variables el nombre de cada una de ellas será: base y altura.

*¿Qué tipo de dato debe almacenar cada una de ellas?* Base y altura son distancias, en este caso de una figura geométrica, y sus valores pueden incluir números después del punto decimal, por tanto, el tipo de dato de cada una de las variables es **Decimal**.

*¿Cuántos cálculos necesita resolver con los datos capturados en la pregunta 1?* Para solucionar el problema se necesita hallar el área y el perímetro del rectángulo y para ello es necesario hacer dos operaciones que son:

$$A = B \times H \qquad P = 2(B + H)$$

Como son dos operaciones se requiere de dos variables para almacenar los resultados, en una de ellas el área y en la otra el perímetro. Los nombres para cada una de las variables serán: área y perímetro.

*¿Qué tipo de dato debe guardar cada variable?* Las dos ecuaciones dependen de la base y la altura, por lo que las variables área y perímetro también son de tipo **Decimal**. La declaración de variables queda:

**var**

**Decimal** : base, altura, área, perímetro

En las ecuaciones a resolver el único valor que es constante durante la ejecución del programa es el número 2 que se encuentra en la siguiente expresión:  $P = 2(B + H)$ .

Se puede declarar como constante así:

```
const  
C1 = 2
```

Por último, se procede a transformar el algoritmo para construir el bloque de ejecución. El paso 2. del algoritmo requiere la capturar datos, lo que corresponde a la instrucción **Leer** en pseudocódigo.

**Leer** (base, altura)

Se acompaña la instrucción **Leer** de una instrucción **Escribir**, esto con el fin de informarle al usuario lo que se desea que éste ingrese. Entonces las dos primeras instrucciones después de **Inicio** son:

```
Escribir ("Ingrese la base y la altura del rectángulo")  
Leer (base, altura)
```

El paso 3. implica la solución de las ecuaciones. Estas ecuaciones deben ser convertidas a expresiones algorítmicas para poder ser usadas dentro del programa. Teniendo en cuenta que las letras A, B, P y H se reemplazan por las variables área, base, perímetro y altura respectivamente, entonces las ecuaciones quedan:

```
area ← base * altura  
perimetro ← 2 * (base + altura)
```

Como se declaró la constante C1 entonces la instrucción de asignación para hallar el perímetro queda:

```
perimetro ← C1 * (base + altura)
```

En el 4. paso se debe visualizar los resultados del área y el perímetro y para ello se utiliza la instrucción **Escribir**:

**Escribir** (“El área y perímetro del rectángulo son:”, area, perimetro)

Entonces el pseudocódigo quedaría:

```

Algoritmo_calculo_area_y_perimetro_rectangulo ← Cabecera
var
    Decimal : base, altura, perímetro, area
const
    C1 = 2
Inicio
    Escribir (“ ingrese base y altura del rectángulo “)
    Leer (base, altura)
    area ← base * altura
    perímetro ← C1 * ( base + altura)
    Escribir (“El área y perímetro del rectángulo son:”,
    area, perimetro)
Fin
    
```

} **Bloque de Declaración**

} **Bloque de Ejecución**

Comparando el algoritmo con el pseudocódigo tenemos:

<b>Algoritmo</b>	<b>Pseudocódigo</b>
1. Inicio	Algoritmo_calculo_area_y_perimetro_rectangulo
2. Capturar la base y altura del rectángulo	<b>var</b>
	Decimal : base, altura, perímetro, area
	<b>const</b>
	C1 = 2
Resolver las ecuaciones del área y	→ <b>Inicio</b>
Perímetro del rectángulo	→ { <b>Escribir</b> (“Ingrese base y altura del rectángulo”)
$A = B \times H$ $P = 2(B + H)$	→ { <b>Leer</b> (base, altura)
Mostrar los resultados del área y	→ { area ← base * altura
perímetro del rectángulo	→ { perímetro ← C1 * ( base + altura)
5. Fin	→ <b>Escribir</b> (“El área y perímetro del rectángulo son:”, area, perimetro)
	→ <b>Fin</b>



Todo Diagrama de flujo comienza y termina con los símbolos de Inicio y Fin, no existe ningún símbolo para representar la cabecera ni la zona de declaración. Inicialmente se construirá el diagrama de flujo partiendo del pseudocódigo y, luego se hará lo mismo pero esta vez usando el algoritmo, ver Ilustración 2

### Pseudocódigo

Algoritmo\_calculo\_area\_y\_perimetro\_rectangulo

**var**

Decimal : base, altura, perímetro, área

**const**

C1 = 2

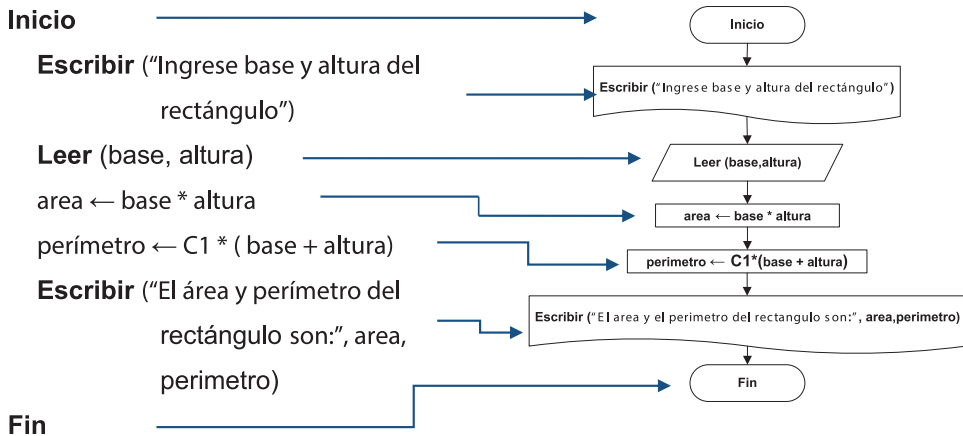


Ilustración 2. Relación entre el pseudocódigo y el diagrama de flujo

En forma general, realizar un Diagrama de flujo es muy sencillo tan solo se debe tener en cuenta las instrucciones y los símbolos asociadas a éstas así como también, el uso de conectores que los unan. La Ilustración 3 muestra cómo se obtiene el Diagrama de flujo a partir del Algoritmo.

### Algoritmo

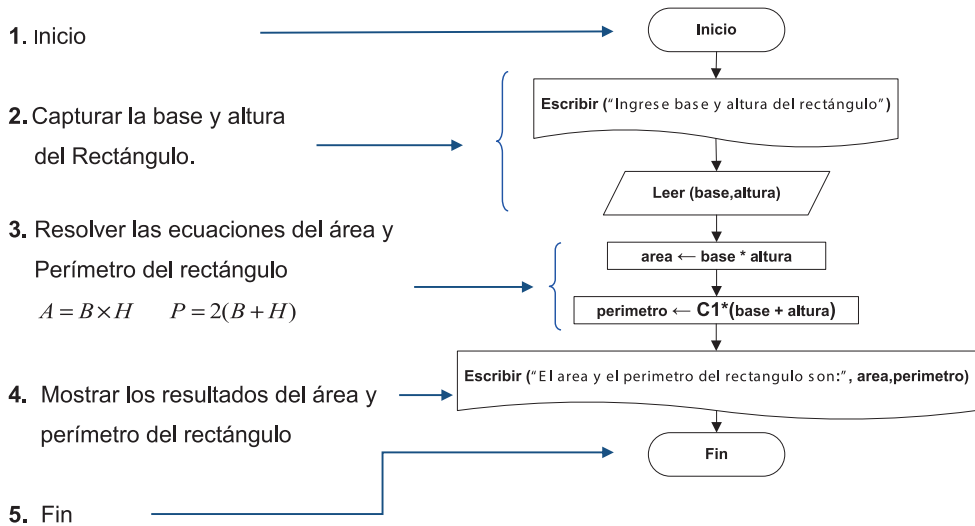


Ilustración 3. Relación entre el algoritmo y el diagrama de flujo.

5. Realice un algoritmo, pseudocódigo y diagrama de flujo tal que, dado el nombre de una persona (de máximo 8 caracteres), el apellido (de máximo 6), la edad en años y su peso en libras, calcule y muestre la edad en número de días, el peso en kilogramos, y el apellido seguido del nombre.

Lo primero es conceptualizar el mundo del problema. Es necesario conocer los datos de nombre, apellido, peso y edad de una persona. Los últimos dos datos son importantes porque con ellos se debe llevar a cabo una conversión. Es necesario visualizar el apellido y luego el nombre contrario a la manera en que se capturan estos datos.

El objetivo es encontrar y visualizar la edad en días, el peso en kilos así como también mostrar el apellido y el nombre de la persona. Para obtener la edad en días y el peso en kilos es necesario llevar a cabo una conversión, es decir:

$$Edad \text{ en días} = Edad \text{ en años} \times \frac{365 \text{ días}}{1 \text{ año}} \Rightarrow Edad \text{ en días} = Edad \text{ en años} \times 365$$

$$Peso \text{ en kilos} = Peso \text{ en libras} \times \frac{1 \text{ kilo}}{2 \text{ libras}} \Rightarrow Peso \text{ en kilos} = \frac{Peso \text{ en libras}}{2}$$

Ahora se construye el algoritmo:

1. Inicio.
2. Capturar el nombre, apellido, edad en años y peso en libras de una persona.
3. Convertir los años a días y las libras a kilos. Para obtener estos datos se utilizan las siguientes ecuaciones:

$$\text{Edad en días} = \text{Edad en años} \times 365$$

$$\text{Peso en kilos} = \frac{\text{Peso en libras}}{2}$$

4. Visualizar la edad en número de días, el peso en kilos, el apellido y el nombre de la persona.
5. Fin.

Ahora se construye el pseudocódigo a partir del algoritmo. En el paso **2.** capturar implica el uso de dos instrucciones, **Escribir** y **Leer**. La palabra convertir, del paso **3.**, está asociada a la instrucción de asignación ya que por medio de ella se lleva a cabo el proceso de transformación de los datos ingresados por el usuario. Visualizar se realiza utilizando la instrucción **Escribir**.

Usando la **Técnica de las Preguntas** se tiene lo siguiente:

*¿Cuántos datos necesita conocer para resolver el problema?* Resolver el problema implica conocer el nombre, apellido, edad en años y peso en libras de una persona, sin estos datos el programador no podrá encontrar la solución. Es necesario capturar 4 datos por lo que se requieren cuatro variables para almacenarlos. Los identificadores de las cuatro variables son: nom, ape, edad\_años y peso\_libras respectivamente. Una vez conocido los nombres de las variables es necesario saber el tipo de las mismas; nom y ape son variables que guardarán el nombre (de 8 caracteres) y el apellido (máximo 6 caracteres) de la persona respectivamente, y como contienen más de una letra el tipo de dato para ellas es **Cadena**. La edad de una persona generalmente es un número que no incluye decimales por tanto el tipo de dato para edad\_años es **Entero**. El peso de una persona puede contener números después del punto decimal por lo que la variable peso\_libras es de tipo **Decimal**.

¿Cuántos cálculos se necesitan resolver con los datos capturados en la pregunta 1? Para resolver el ejercicio es necesario convertir la edad de años a días y el peso de libras a kilos, por lo que se requieren de dos operaciones y de dos variables que almacenen los resultados. Los identificadores para estas dos variables son: edad\_días y peso\_kilos respectivamente. Generalmente cuando se habla de cantidad de días se hace referencia a un número entero por lo que el tipo de dato para edad\_días es **Entero**. El peso de una persona puede contener números después del punto decimal por lo que la variable peso\_kilos es de tipo **Decimal**.

Con base en lo anterior la zona de declaración de variables queda:

```
var
    Cadena : nom[8], ape[6]
    Decimal : peso_libras, peso_kilos
    Entero  : edad_años, edad_días
```

Las instrucciones que reflejan los pasos 2., 3. y 4. del algoritmo son:

```
    Escribir (“Ingrese el nombre, apellido, edad en años y peso en libras de la persona”)
    Leer (nom, ape, edad_años, peso_libras)
    edad_días ← edad_años * 365
    peso_kilos ← peso_libras / 2
    Escribir (“La edad en días, el peso en kilos:”, edad_días, peso_kilos)
    Escribir (“el apellido y nombre de la persona son:”, ape, nom)
```

A continuación la Ilustración 4 muestra el pseudocódigo completo y el correspondiente diagrama de flujo:

### **Pseudocódigo**

Algoritmo\_calculo\_potencia\_circuito\_electrico

**Var**

```
    Cadena : nom[8], ape[6]
    Decimal : peso_libras, peso_kilos
    Entero  : edad_años, edad_días
```

**Inicio**

**Escribir** (“ingrese el nombre, apellido, edad en años y peso en libras de la persona”)

**Leer** (nom, ape, edad\_años, peso\_libras)

$edad\_días \leftarrow edad\_años * 365$

$peso\_kilos \leftarrow peso\_libras / 2$

**Escribir** (“La edad en días, el peso en kilos:”, edad\_días, peso\_kilos)

**Escribir** (“el apellido y nombre de la persona son:”, ape, nom)

**Fin**

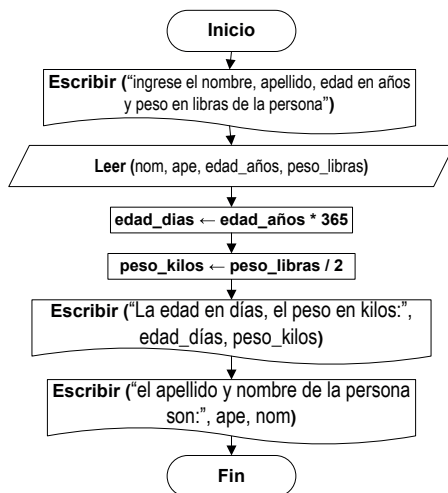


Ilustración 4. Relación entre el pseudocódigo y el diagrama de flujo.

6. La potencia que consume un bombillo se expresa por medio de la fórmula  $P = V \times I$ , donde  $V$  es el voltaje e  $I$  es la corriente. El voltaje está en función de la corriente  $I$  y la resistencia  $R$  del bombillo, por medio de la siguiente expresión:  $V = I \times R$ . La potencia total entregada por la fuente es igual a la suma de las potencias de todos los bombillos. La figura siguiente muestra cuatro (4) bombillos, y ellos están conectados a una fuente. Construya un algoritmo, pseudocódigo y diagrama de flujo tal que, conocidos el valor de la corriente  $I$  y los valores de resistencia  $R$  - de cada bombillo -, calcule y muestre la potencia de cada bombillo y la potencia total entregada por la fuente, ver Ilustración 5.

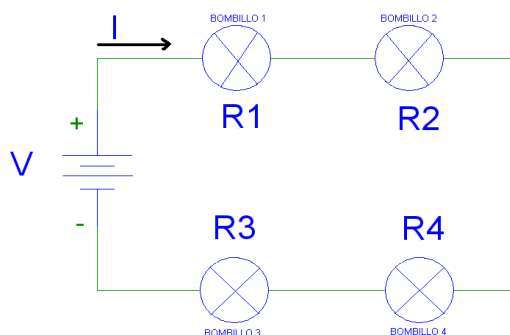


Ilustración 5. Circuito serie conformado por cuatro bombillos y una fuente de voltaje.

Conceptualización: el circuito contiene cuatro bombillos representados con las palabras R1, R2, R3 y R4 (la letra R es de resistencia) conectados a una fuente de voltaje V. La figura muestra que a los cuatro bombillos los atraviesa la corriente I, con un valor igual para cada uno de ellos y a su vez, está sale y retorna a la fuente V. Según el enunciado, multiplicar la corriente I por el valor de la resistencia de un bombillo produce el voltaje de ese bombillo en particular, además, multiplicar este voltaje por el valor de la corriente genera la potencia que consume el bombillo.

Objetivo: encontrar la potencia de cada bombillo así como también la potencia total entregada por la fuente y para ello, el enunciado establece que se conocen la corriente I que atraviesa el circuito y los valores de cada resistencia. Para calcular la potencia de cada bombillo se hace uso de

$$P = V \times I$$

En donde V es igual a:

$$V = I \times R$$

Se puede sustituir el valor de V en la primera ecuación por el que éste tiene en la segunda expresión, entonces

$$P = V \times I = (I \times R) \times R = I^2 \times R$$

Ahora se construye el algoritmo:

1. Inicio.
2. Capturar el valor de la corriente I y los valores de las resistencias **R1**, **R2**, **R3** y **R4** del circuito.
3. Calcular la potencia de cada bombillo y la potencia total de la fuente del circuito. La potencia de cada bombillo se puede obtener por el uso de

$$P = I^2 \times R$$

Donde se reemplaza el valor de I y R por el capturado en el punto 2, note que se deben hallar cuatro valores de potencia y para cada uno de ellos cambia el valor de R quedando igual el valor de I.

La potencia total se puede hallar de la siguiente manera:

$$P = P1 + P2 + P3 + P4$$

Donde P1, P2, P3 y P4 son las potencias del bombillo R1, R2, R3 y R4 respectivamente.

4. Mostrar la potencia de cada bombillo así como la potencia total del circuito.
5. Fin.

Con base en el algoritmo se construirá el pseudocódigo y para ello se hará uso de las palabras claves escritas en aquel con el fin de elaborar las instrucciones pertinentes. El nombre que se le dará a este programa será cálculo área y perímetro rectángulo, por tanto la **cabecera** queda:

Algoritmo\_calculo\_potencia\_circuito\_electrico

Para determinar cuántas variables y de qué tipo son necesarias declarar se usará en este caso la técnica del algoritmo, es decir:

*El punto dos nos informa de cuántas variables se necesitan crear para poder almacenar los datos que se desean capturar.* En este caso se necesitan de cinco variables, una para la corriente y las otras cuatro para los valores de las resistencias de los bombillos, a cada una de ellas se les asignan los siguientes identificadores **I**, **R1**, **R2**, **R3** y **R4** respectivamente. *¿Qué tipo de dato se guardará en I, R1, R2, R3 y R4?* Los equipos eléctricos que miden estas variables registran valores reales por lo que cada una de estas variables debe ser de tipo **Decimal**, entonces hasta el momento la **Declaración de variables** tiene:

**var**

**Decimal** : I, R1, R2, R3, R4

*El punto tres muestra cuántos cálculos se deben resolver,* en este caso se deben llevar a cabo cinco cálculos, cuatro para las potencias de cada bombillo y uno para la potencia total, por tanto, es necesario declarar cinco variables a cada una de ellas se les asignan los siguientes identificadores **P1**, **P2**, **P3**, **P4** y **Pt**, el tipo de dato que éstas deben almacenar es **Decimal**.

La declaración de variables usando la técnica del algoritmo es:

**var**

**Decimal** : I, R1, R2, R3, R4, P1, P2, P3, P4, Pt

Aunque la ecuación del cálculo de la potencia de cada bombillo existe el número dos, no se construirá la solución del ejemplo utilizando la declaración de constantes.

El punto dos del algoritmo contiene la palabra clave *Capturar*, la cual indica que se debe hacer uso de la instrucción **Leer**, pero la misma debe estar acompañada de la instrucción **Escribir**. Dentro de la instrucción **Leer** se deben colocar las variables que almacenaran los datos que se necesitan capturar y, se utiliza la instrucción **Escribir** con el fin de informar al usuario qué datos debe ingresar los cuales serán capturados cuando se ejecute la instrucción **Leer**. Las instrucciones quedan:

**Escribir** (“ingrese la corriente I y los valores de R1, R2, R3 y R4 del circuito”)

**Leer** ( I, R1, R2, R3, R4)

El punto tres del algoritmo informa sobre cuáles y cuántos cálculos se deben realizar para resolver el problema, este punto está relacionado con la instrucción de asignación. Las instrucciones quedan:

#### Aritmética

$$P1 = I^2 \times R1$$

$$P2 = I^2 \times R2$$

$$P3 = I^2 \times R3$$

$$P4 = I^2 \times R4$$

$$Pt = P1 + P2 + P3 + P4$$

#### Algorítmica

$$P1 \leftarrow I^{**2} * R1$$

$$P2 \leftarrow I^{**2} * R2$$

$$P3 \leftarrow I^{**2} * R3$$

$$P4 \leftarrow I^{**2} * R4$$

$$Pt \leftarrow P1 + P2 + P3 + P4$$

El punto cuatro del algoritmo permite la visualización de los resultados, para llevar a cabo esto es necesario el uso de la instrucción **Escribir**, las instrucciones quedan:



**Escribir** (“La potencia de los bombillos 1, 2, 3 y 4 del circuito es:”, P1, P2, P3, P4)

**Escribir** (“La potencia total de la fuente es:”, Pt)

El último punto del algoritmo es la terminación del mismo y para ello se utiliza la palabra reservada **Fin**.

A continuación se muestra el pseudocódigo junto al algoritmo y se hace énfasis en la relación entre ellos usando flechas las cuales ayudaran al lector a observar la correspondencia entre el algoritmo y el pseudocódigo. Al igual que la relación entre el pseudocódigo y el diagrama de flujo, ver Ilustración 6.

#### Algoritmo

1. Inicio

2. Capturar el valor de la corriente **I** y los valores de las resistencias **R1**, **R2**, **R3** y **R4** del circuito eléctrico.

3. Calcular la potencia de cada bombillo y la potencia total de la fuente del circuito eléctrico.

$$P1 = I^2 \times R1 \quad P2 = I^2 \times R2$$

$$P3 = I^2 \times R3 \quad P4 = I^2 \times R4$$

$$Pt = P1 + P2 + P3 + P4$$

4. Mostrar la potencia de cada bombillo así como la potencia total del circuito eléctrico.

5. Fin

#### Pseudocódigo

Algoritmo\_calculo\_potencia\_circuito\_electrico

**var**

**Decimal:** I, R1, R2, R3, R4, P1, P2, P3, P4, Pt

→ **Inicio**

→ { **Escribir** (“ingrese la corriente **I** y los valores de **R1**, **R2**, **R3** y **R4** del circuito eléctrico”)

**Leer** ( I, R1, R2, R3, R4)

→ {  $P1 \leftarrow I^2 * R1$   
 $P2 \leftarrow I^2 * R2$   
 $P3 \leftarrow I^2 * R3$   
 $P4 \leftarrow I^2 * R4$   
 $Pt \leftarrow P1 + P2 + P3 + P4$

→ { **Escribir** (“La potencia de los bombillos 1, 2, 3 y 4 del circuito eléctrico es:”, P1, P2, P3, P4)

**Escribir** (“La potencia total de la fuente es:”, Pt)

→ **Fin**

### **ALGORITMO usando ESC**

Inicio  
Capturar tres números enteros positivos.  
Si (  $A \geq 0$  Y  $B \geq 0$  Y  $C \geq 0$  ) entonces  
Si (  $A <> B$  Y  $A <> C$  Y  $B <> C$  ) entonces  
Si (  $A > B$  Y  $A > C$  ) entonces  
    Mostrar el mayor es A  
De lo contrario  
    Si (  $B > A$  Y  $B > C$  ) entonces  
        Mostrar el mayor es B  
De lo contrario  
        Mostrar el mayor es C  
Fin\_si  
Fin\_si  
De lo contrario  
    Mostrar existen números iguales  
Fin\_si  
De lo contrario  
    Mostrar numero no valido  
Fin\_si  
Fin

### **Pseudocódigo usando ESC**

Algoritmo\_encuentra\_mayor  
var  
    Entero : A,B,C  
Inicio  
    Escribir("Ingrese tres números enteros positivos  
        diferentes")  
    Leer (A,B,C)  
    **Si** (  $A \geq 0$  Y  $B \geq 0$  Y  $C \geq 0$  ) **entonces**  
        **Si** (  $A <> B$  Y  $A <> C$  Y  $B <> C$  ) **entonces**  
            **Si** (  $A > B$  Y  $A > C$  ) **entonces**  
                Escribir ("El numero mayor es:",A)  
            **De lo contrario**  
                **Si** (  $B > A$  Y  $B > C$  ) **entonces**  
                    Escribir ("El numero mayor es:",B)  
                **De lo contrario**  
                    Escribir ("El numero mayor es:",C)  
            **Fin\_si**  
        **Fin\_si**  
    **De lo contrario**  
        Escribir ("Existen números iguales")  
    **Fin\_si**  
    **De lo contrario**  
        Escribir ("numero no valido")  
    **Fin\_si**  
Fin

Ilustración 6. Relación entre el pseudocódigo y el diagrama de flujo.

7. En la tienda Wall Max existen los siguientes productos:

Artículo	Marca	Valor US\$
Camisa	<i>T-shirt</i>	\$15
Pantalón	<i>Lave`se</i>	\$25 el par
Zapatos	<i>Acme</i>	\$80
Ropa interior	<i>Proteggeme</i>	\$45
Medias	<i>Ropita pa pie</i>	\$10
DVD	<i>Pollito</i>	\$70

La tienda ofrece un descuento del: 10% en las camisas y 10% en los pantalones por la compra de 3 de cada uno de ellos; 4% por la compra de dos pares de zapatos y el 50% de descuento por el tercer par; por la compra de cinco pares de medias el descuento es del 5% y 8% de descuento por comprar dos DVD. Construya un algoritmo, pseudocódigo y diagrama de flujo que muestre: el valor total de cada artículo que se compra, el valor de cada descuento obtenido, el valor total a pagar.

Conceptualización: como en toda tienda, los clientes pueden o no comprar uno más productos y varios de cualquier artículo. El precio final a pagar depende de la cantidad que compre el cliente de cada artículo, por ejemplo, si se compran dos camisas, éstas tendrán un precio de US \$30 dólares pero si se llevan 3, el valor final a pagar debe tener incluido el descuento del 10%. El único producto que no tiene descuento es la ropa interior. Un solo pantalón cuesta US \$12.5 dólares.

Objetivo: encontrar y visualizar el valor de cada artículo que se compra, por ejemplo, un pantalón vale US \$12.5 dólares, por dos son US \$25 dólares. Además se debe obtener y mostrar el valor de cada descuento obtenido así como el valor total a pagar.

Obtener el descuento implica que el programa pueda reconocer la cantidad de productos que se compran de un artículo basándose en las características del mismo expresadas en el enunciado, pero para hallar este valor es necesario construir las ecuaciones pertinentes y para ello se presenta el siguiente análisis:

El enunciado establece que por la compra de 3 productos del artículo camisas o pantalones tiene un descuento del 10%, pero qué pasa cuando se compra 1 o dos camisas o incluso 6 o 9 de ellas? Si se compra una camisa el valor de este producto es de US \$15 dólares; dos valen US \$30 dólares; para 3 se debe aplicar el descuento del 10% (10% = 10/100 o 0.1), es decir:

$$\begin{array}{rcl}
 3 \times 15 & = & 45 \quad \text{Valor a pagar sin descuento} \\
 3 \times 15 \times 10 / 100 & = & 4.5 \quad \text{Valor a pagar del descuento} \\
 3 \times 15 - 3 \times 15 \times 10 / 100 & = & 40.5 \quad \text{Valor a pagar incluido el descuento}
 \end{array}$$

Por las tres camisas se debe pagar un total de US \$40.5, este valor es equivalente a:

$$3 \times 15 - 3 \times 15 \times 0.1 = 3 \times 15 (1 - 0.1) = 3 \times 15 \times 0.9 = 40.5$$

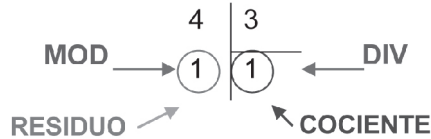
Donde 0.9 es igual a 90% que es el resultado de restarle al 100% el 10%. La expresión anterior permite calcular el valor a pagar por tres camisas incluido el descuento. En la expresión anterior los números 3 y 15 representan la cantidad de camisas y su precio por unidad respectivamente.

Cuando se compran 4 camisas, cómo se calcula el valor a pagar por esta cantidad? El programa construido debe detectar si en la cantidad de camisas compradas existen múltiplos o grupos de tres para aplicarles el descuento que ofrece la tienda; el número 4 se puede descomponer en 3 y 1, por lo que se aprecia que al llevar 4 camisas tres de éstas tienen el descuento del 10% y la cuarta no, es decir:

<b>Cantidad</b>	<b>Valor a pagar dólares</b>
4 → 3 →	$3 \times 15 \times 0.9 = 40.5$
→ 1 →	$1 \times 15 = 15$
<b>Total</b>	$3 \times 15 \times 0.9 + 1 \times 15 = 55.5$

Entonces por cuatro camisas el cliente debe pagar \$40.5 dólares.

Pero, cómo se puede obtener de un 4 un 3 y un 1? Como se busca encontrar cuantos grupos de 3 existen dentro de una cantidad determinada de camisas, se debe utilizar una operación aritmética que permita conseguir estos múltiplos de un valor dado de camisas y, la mejor manera para lograr esto es usando la división.



En la división anterior se muestra que el número 3 es el divisor y el 4, la cantidad de camisas, es el dividendo. Note que se obtiene un residuo y cociente de valor 1. Sin embargo, cómo se interpretan los resultados? El residuo, como en toda división, indica lo que queda, el resto, y en este ejemplo significa una camisa y al multiplicar el residuo por el precio de este artículo se obtiene:

$$1 \times 15 = 15 \text{ dólares}$$

Que es el valor de una unidad del artículo camisa. Para obtener el residuo se hace uso del operador **MOD** y se construye la siguiente expresión:

$$4 \text{ MOD } 3 = 1$$

Y la expresión que encuentra el valor a pagar por la cantidad de camisas que representan el residuo es:

$$4 \text{ MOD } 3 * 15$$

Esta ecuación genera como resultado 15 que representa la cantidad de dinero que debe cancelar el cliente por la compra de una unidad del artículo camisa pero aún falta cobrar por las tres restantes. La expresión anterior permite encontrar el valor a pagar por una cantidad de camisas inferior a 3.

El resultado en el cociente representa la cantidad de grupos según el valor del divisor, es decir, cuatro dividido 3 genera un cociente de 1 que significa un grupo de 3, para el caso particular un grupo de 3 camisas y para éstas la tienda ofrece un descuento del 10%. El cociente se obtiene utilizando el operador **DIV** de la siguiente manera:

$$4 \text{ DIV } 3 = 1$$

Que representa un grupo de 3. Al multiplicar este valor por el divisor se obtiene la cantidad total de camisas que conforma el grupo, es decir:

$$4 \text{ DIV } 3 * 3 = 3$$

Por esta cantidad la tienda ofrece un descuento del 10%, entonces para obtener el valor final a pagar por estas tres camisas se multiplica el resultado anterior por el precio y por 0.9, es decir:

$$4 \text{ DIV } 3 * 3 * 15 * 0.9 = 40.5$$

Que representa el mismo valor que se obtuvo por la compra de 3 camisas incluido el descuento. El valor total a pagar por las cuatro camisas se obtiene de sumar:

$$4 \text{ MOD } 3 * 15$$

con

$$4 \text{ DIV } 3 * 3 * 15 * 0.9$$

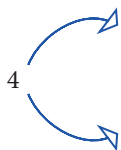
Es decir,

$$4 \text{ MOD } 3 * 15 + 4 \text{ DIV } 3 * 3 * 15 * 0.9 = 55.5$$

Para cantidades superiores o iguales a 3 se observa en la Tabla 11:

**Tabla 11. Análisis de las cantidades de camisas para el ejemplo 4.**

Cantidad	Valor a pagar dólares	
3	$3 \times 15 \times 0.9$	= 40.5
3	$3 \times 15 \times 0.9$	= 40.5
1	$1 \times 15$	= 15
<b>Total</b>	$3 \times 15 \times 0.9 + 1 \times 15$	= 55.5



5	3	$3 \times 15 \times 0.9$	$= 40.5$
	2	$2 \times 15$	$= 30$
	<b>Total</b>	$3 \times 15 \times 0.9 + 2 \times 15$	$= 70.5$
6	3	$3 \times 15 \times 0.9$	$= 40.5$
	3	$3 \times 15 \times 0.9$	$= 40.5$
	<b>Total</b>	$3 \times 15 \times 0.9 + 3 \times 15 \times 0.9$	$= 81$
7	6	$6 \times 15 \times 0.9$	$= 81$
	1	$1 \times 15$	$= 15$
	<b>Total</b>	$3 \times 15 \times 0.9 + 2 \times 15$	$= 96$

Fuente: Autores

En la Tabla 11 que para una cantidad superior o igual a 3 se descompone ésta en dos valores y uno de ellos, el del múltiplo de 3, está presente en todas las expresiones que determinan el valor total a pagar por la compra.

Para una cantidad de cinco camisas y utilizando el análisis previo se tiene que la ecuación que permite encontrar el valor a total a pagar es:

$$5 \text{ MOD } 3 * 15 + 5 \text{ DIV } 3 * 3 * 15 * 0.9 = 70.5$$

La única diferencia entre las ecuaciones que encuentran el valor total a pagar para 4 y 5 camisas, es el valor que representa la cantidad, es decir, el 4 o el 5, y es exactamente este dato el que siempre está cambiando y es ingresado por el usuario, entonces para generalizar la expresión que permite hallar el valor final a pagar incluido el descuento por la compra de camisas es:

$$V_{tpc} \leftarrow CC \text{ MOD } 3 * 15 + CC \text{ DIV } 3 * 3 * 15 * 0.9$$

Donde CC es la variable que almacena el valor de la cantidad de camisas ingresado por el usuario y V<sub>tpc</sub> es la variable que almacena el valor total a pagar

por las camisas. La siguiente tabla muestra la comprobación de la ecuación anterior.

**Tabla 12. Comprobación de la ecuación  $V_{tpc}$  para el ejemplo 4.**

Cantidad	Primer término		Segundo término		$V_{tpc}$
	CC MOD 3 * 15	Resultado	CC DIV 3 * 3 * 15 * 0.9	Resultado	
0	0 MOD 3 * 15	0	0 DIV 3 * 3 * 15 * 0.9	0	0
1	1 MOD 3 * 15	15	1 DIV 3 * 3 * 15 * 0.9	0	15
2	2 MOD 3 * 15	30	2 DIV 3 * 3 * 15 * 0.9	0	30
3	3 MOD 3 * 15	0	3 DIV 3 * 3 * 15 * 0.9	40.5	40.5
4	4 MOD 3 * 15	15	4 DIV 3 * 3 * 15 * 0.9	40.5	55.5
5	5 MOD 3 * 15	30	5 DIV 3 * 3 * 15 * 0.9	40.5	70.5
6	6 MOD 3 * 15	0	6 DIV 3 * 3 * 15 * 0.9	81	81

Fuente: Autores

Los resultados obtenidos con la ecuación de  $V_{tpc}$  coinciden con los de la Tabla 12 así como con los del análisis previo de este ejemplo. Obsérvese como el término que incluye el operador **MOD** permite obtener el precio a pagar para una cantidad de 1 o 2 productos sea que éstos representen la cuantía de la compra ó los artículos que “sobran” de aquellos que tienen descuento (por ejemplo, de 4 artículos uno de ellos no tiene descuento).

La ecuación general del descuento para las camisas es:

$$dtoc \leftarrow CC \text{ DIV } 3 * 3 * 15 * 0.1$$

Aunque el análisis anterior se realizó para las camisas, para los pantalones la tienda Wall Mart ofrece el mismo tipo de descuento por lo que utilizando los mismos criterios se puede concluir que la expresión que permite obtener el valor total a pagar por la compra de pantalones es:

$$V_{tpp} \leftarrow CP \text{ MOD } 3 * 12.5 + CP \text{ DIV } 3 * 3 * 12.5 * 0.9$$

Esta expresión se diferencia de  $V_{tpc}$  en el precio unitario por pantalón.  $V_{tpp}$  y  $CP$  son las variables que almacenan el valor total a pagar por pantalones y la cantidad de pantalones respectivamente.

Para los artículos medias y DVD las expresiones son las siguientes:

$$V_{tpm} \leftarrow CM \text{ MOD } 5 * 10 + CM \text{ DIV } 5 * 5 * 10 * 0.95$$

$$V_{tpdvd} \leftarrow CDVD \text{ MOD } 2 * 70 + CDVD \text{ DIV } 2 * 2 * 70 * 0.92$$



En estas ecuaciones  $V_{tpm}$ ,  $V_{tpdvd}$ ,  $CM$  y  $CDVD$  son valor total a pagar medias, valor total a pagar dvd, cantidad de medias y cantidad de dvd respectivamente. En estas expresiones que se usa 0.95 y 0.92 para representar el 5% y 8% respectivamente además, obsérvese que el divisor para  $V_{tpm}$  es el 5 y para  $V_{tpdvd}$  el 2 que representan la cantidad de productos de medias y dvd para los cuales la tienda ofrece descuento.

Las ecuaciones de descuento para las camisas, pantalones, medias y dvd son:

$$\begin{aligned} dtoc &\leftarrow CC \text{ DIV } 3 * 3 * 15 * 0.1 \\ dtop &\leftarrow CP \text{ DIV } 3 * 3 * 12.5 * 0.1 \\ dtom &\leftarrow CM \text{ DIV } 5 * 5 * 10 * 0.05 \\ dtodvd &\leftarrow CDVD \text{ DIV } 2 * 2 * 70 * 0.08 \end{aligned}$$

La tienda ofrece un descuento especial para el artículo *Ropita pa pie*, 4% por la compra de dos pares de zapatos y el 50% de descuento por el tercer par y el programa construido debe entregar el valor a pagar por cualquier cantidad comprada de este artículo. El siguiente análisis muestra como se obtienen los términos de la expresión con la cual se halla el valor total a pagar por zapatos.

Si se compra un par de zapatos el valor de este producto es de US \$70 dólares; para dos pares se debe aplicar el descuento del 4% ( $4\% = 4/100$  o 0.04), es decir:

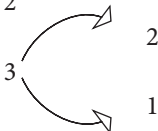
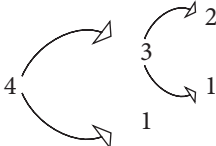
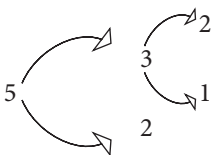
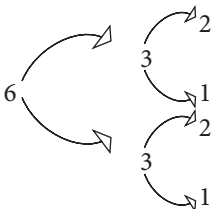
$$\begin{aligned} 2 \times 70 &= 140 && \text{Valor a pagar sin descuento} \\ 2 \times 70 \times 4 / 100 &= 5.6 && \text{Valor a pagar del descuento} \\ 2 \times 70 - 2 \times 70 \times 4 / 100 &= 134.4 && \text{Valor a pagar incluido el descuento} \end{aligned}$$

Para tres pares la expresión debe incluir el 50% ( $50\% = 50/100$  o 0.5) de descuento para el tercer par y el 4% para los otros dos pares, es decir

$$\begin{aligned} 2 \times 70 &= 140 && \text{Valor a pagar sin descuento} \\ 2 \times 70 \times 4 / 100 &= 5.6 && \text{Valor a pagar del descuento} \\ 2 \times 70 - 2 \times 70 \times 4 / 100 &= 134.4 && \text{Valor a pagar incluido el descuento} \\ 1 \times 70 \times 0.5 &= 35 && \text{Valor del tercer par} \\ 2 \times 70 \times 0.96 + 1 \times 70 \times 0.5 &= 169.4 && \text{Valor a pagar incluido el descuento} \end{aligned}$$

La Tabla 13 muestra cómo se discrimina el cobro según la cantidad de zapatos comprados.

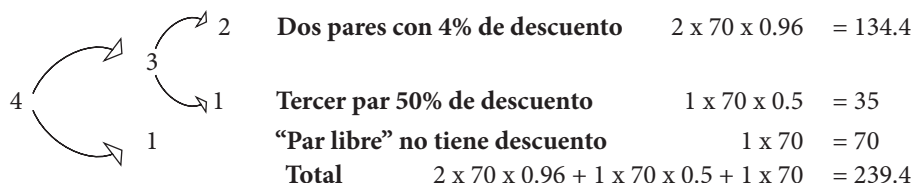
**Tabla 13. Análisis de las cantidades de zapatos para el ejemplo 4.**

Cantidad	Valor a pagar dólares
1	$1 \times 70 = 70$
2	$2 \times 70 \times 0.96 = 134.4$
 3 2 1	$2 \times 70 \times 0.96 = 134.4$ $1 \times 70 \times 0.5 = 35$
<b>Total</b>	$2 \times 70 \times 0.96 + 1 \times 70 \times 0.5 = 169.4$
 4 3 2 1	$2 \times 70 \times 0.96 = 134.4$ $1 \times 70 \times 0.5 = 35$
<b>Total</b>	$2 \times 70 \times 0.96 + 1 \times 70 \times 0.5 + 1 \times 70 = 239.4$
 5 3 2 1	$2 \times 70 \times 0.96 = 134.4$ $1 \times 70 \times 0.5 = 35$
<b>Total</b>	$2 \times 70 \times 0.96 + 1 \times 70 \times 0.5 + 2 \times 70 \times 0.96 = 303.8$
 6 3 2 3 2 1	$2 \times 70 \times 0.96 = 134.4$ $1 \times 70 \times 0.5 = 35$ $2 \times 70 \times 0.96 = 134.4$
<b>Total</b>	$2 \times (2 \times 70 \times 0.96) + 2 \times (1 \times 70 \times 0.5) = 338.4$

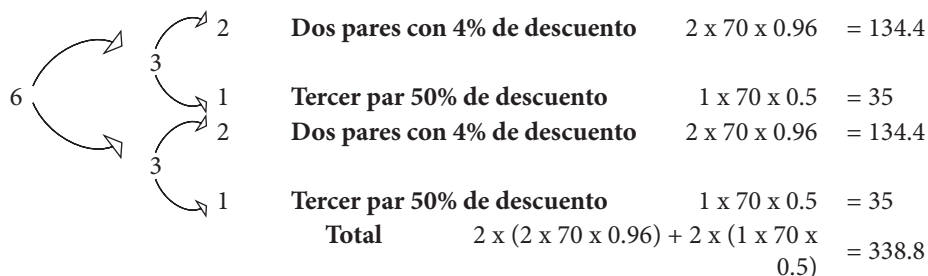
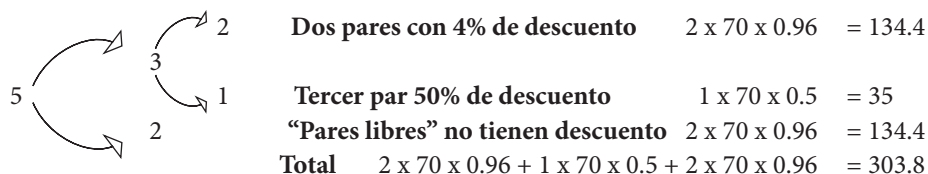
El valor a pagar por dos pares de zapatos incluido el descuento se puede representar de la siguiente manera:

$$2 \times 70 - 2 \times 70 \times 4 / 100 = 2 \times 70 (1 - 0.04) = 2 \times 70 \times 0.96$$

De la Tabla 13 que para una cantidad de 3 o más existe en común grupos de 3 pares entre los cuales se encuentra el tercer par que tiene el descuento del 50%, los otros dos pares son cobijados por el descuento del 4% de la tienda. Por ejemplo, cuatro pares se pueden descomponer en 3 y 1, es decir:

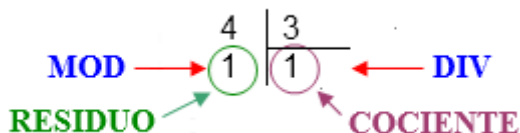


Para cinco y seis pares se tiene lo siguiente:



En todos los casos de cualquier cantidad de pares de zapatos, la misma se puede descomponer en grupos de 3, pero por qué este valor? Se escoge este número debido al descuento del 50% del tercer par, por lo que siempre que se encuentre un grupo de tres se puede garantizar que dentro de éste se halla el par con el 50% de descuento y los pares restantes dentro de este grupo tendrán el 4%, esto significa entonces que se debe construir una expresión que permita obtener el 3 para así generar el valor a pagar por esta cantidad.

Para obtener un 3 y un 1 de un cuatro se realiza el siguiente procedimiento:



Siempre que el **DIV** genere un valor diferente a cero, se asegura que existe un grupo superior o igual a 3 y con certeza se debe cobrar por dos pares de zapatos incluyendo el 4% de descuento en éstos, y el tercer par a la mitad de precio. La expresión para dos pares es

$$4 \text{ DIV } 3 * 2 * 70 * 0.96 = 134.4$$

Y para el tercer par es

$$4 \text{ DIV } 3 * 70 * 0.5 = 35$$

Según la Tabla 13 para cantidades superiores o iguales a 3 pares de zapatos siempre se puede encontrar grupos de 3 por lo que las dos expresiones anteriores se pueden generalizar para cualquier cantidad de zapatos mayor o igual a 3 pero solo para encontrar el valor a pagar por el grupo, entonces por el grupo de 3 que hacen parte de los cuatro pares el costo a pagar es:

$$4 \text{ DIV } 3 * 2 * 70 * 0.96 + 4 \text{ DIV } 3 * 70 * 0.5 = 169.4$$

Una compra de cinco pares de zapatos, incluye según la Tabla 13 a un grupo de 3 y “dos pares” libres, por tanto, para este grupo de 3 la expresión es

$$5 \text{ DIV } 3 * 2 * 70 * 0.96 + 5 \text{ DIV } 3 * 70 * 0.5 = 169.4$$

Las expresiones para los grupos de 3 que hacen parte de la cantidad 4 o 5 pares de zapatos solo se diferencian en el valor 4 en el primer caso y cinco en el último. Para seis pares existen dos grupos de 3 lo que significa que se repite dos veces las expresiones obtenidas para un grupo de 3, la ecuación para 6 pares es

$$6 \text{ DIV } 3 * 2 * 70 * 0.96 + 6 \text{ DIV } 3 * 70 * 0.5 = 338.4$$

En la expresión anterior el **DIV** genera el 2 logrando esto la duplicidad de cada término en la ecuación.

Entonces para un grupo de 3 se puede generalizar la expresión que permite calcular el precio a pagar por estos tres pares de la siguiente forma:

$$\text{CZ DIV } 3 * 2 * 70 * 0.96 + \text{CZ DIV } 3 * 70 * 0.5$$

Donde CZ representa la variable donde se almacena la cantidad de zapatos ingresada por el usuario.

Hasta el momento la ecuación obtenida permite hallar el valor a pagar por un grupo de 3 en una cantidad de 3 o más pares de zapatos sin embargo aún falta encontrar la expresión para los “pares libres”.

Cuatro pares de zapatos incluyen un “par libre”, en la operación de división el **MOD** muestra que queda un “par libre” el cual no tiene descuento alguno y por el mismo se debe cancelar US \$70 dólares, este valor se obtiene de la siguiente manera

$$4 \text{ MOD } 3 * 70 = 70$$

Para cinco pares de zapatos el **MOD** muestra que quedan dos “pares libres” y, según la oferta de la tienda, para esta cantidad se ofrece un descuento del 4% por lo que estos “pares libres” son cobijados por la promoción, entonces la ecuación que permite obtener el valor a pagar es

$$5 \text{ MOD } 3 * 70 * 0.96 = 134.4$$

De la Tabla 13 que solo se generan dos tipos de valores para los “pares libres” y son 1 y 2, por lo que se necesita una ecuación que permita hallar el valor a pagar por estos residuos y satisfaga para ambas cantidades. Obsérvese que uno de los residuos es par, por lo que se aprovecha esta característica en la ecuación final construyendo dentro de ella el operador que admita solo pares, y para ello se hace uso del **DIV 2**, entonces

$$5 \text{ MOD } 3 \text{ DIV } 2 * 2 * 70 * 0.96 = 134.4$$

$5 \text{ MOD } 3$  da como resultado 2 luego este junto con el **DIV 2** genera 1 que significa un par, este valor se multiplica por 2 obteniéndose el 2, número que indica un par de zapatos a los cuales se les aplica luego el precio y el descuento del 4%. La anterior expresión funciona cuando el **DIV** genera un valor diferente de cero significa entonces que el **MOD** genera un valor par, por tanto, solo sirve para cantidades de zapatos impares y superiores a 3.

Para el valor 1 se cambia el **DIV** por el **MOD** quedando

$$4 \text{ MOD } 3 \text{ MOD } 2 * 70 = 70$$

Esta ecuación trabaja cuando el primer **MOD** genera como resultado un 1, esto indica que funciona para cualquier cantidad de zapatos que satisfaga el primer **MOD**.

Si se generalizan las expresiones para cantidades de zapatos cuyo valor son 1 y 2 se tiene lo siguiente

$$\begin{array}{ll} \text{CZ MOD 3 DIV 2} * 2 * 70 * 0.96 & \text{para el residuo par} \\ \text{CZ MOD 3 MOD 2} * 70 & \text{para residuo 1} \end{array}$$

Entonces la expresión que permite obtener el valor a pagar para cualquier cantidad de zapatos incluido el descuento es

$$V_{tpz} \leftarrow \underbrace{\text{CZ MOD 3 MOD 2} * 70}_{\text{Para residuo 1}} + \underbrace{\text{CZ MOD 3 DIV 2} * 2 * 70 * 0.96}_{\text{Para el residuo par}} + \underbrace{\text{CZ DIV 3} * 2 * 70 * 0.96}_{\text{Para el par dentro de un grupo de 3}} + \underbrace{\text{CZ DIV 3} * 70 * 0.5}_{\text{Para el tercer par 50 \% descuento}}$$

Donde CZ es la variable que almacena el valor de la cantidad de zapatos ingresado por el usuario y Vtpz es la variable que almacena el valor total a pagar por este artículo incluido el descuento. En la construcción de Vtpz se elaboró los términos de la misma partiendo de aquellas cantidades de zapatos que son iguales o superiores a 3 y, que los términos para los residuos en cantidades iguales a 4 o 5 pares satisfacen también para la compra de uno o dos pares de zapatos. La Tabla 14 muestra la comprobación de la ecuación anterior.

La expresión para el descuento de los zapatos es

$$d_{toz} \leftarrow \text{CZ MOD 3 DIV 2} * 2 * 70 * 0.04 + \text{CZ DIV 3} * 2 * 70 * 0.04 + \text{CZ DIV 3} * 70 * 0.5$$

En esta expresión que se incorpora el número 0.04 que corresponde al 4% valor del descuento para dos pares de zapatos.

Las ecuaciones que permiten resolver el problema de la tienda Wall Mart son:

$$\begin{array}{l} V_{tpc} \leftarrow \text{CC MOD 3} * 15 + \text{CC DIV 3} * 3 * 15 * 0.9 \\ V_{tpp} \leftarrow \text{CP MOD 3} * 12.5 + \text{CP DIV 3} * 3 * 12.5 * 0.9 \\ V_{tpm} \leftarrow \text{CM MOD 5} * 10 + \text{CM DIV 5} * 5 * 10 * 0.95 \\ V_{tpdvd} \leftarrow \text{CDVD MOD 2} * 70 + \text{CDVD DIV 2} * 2 * 70 * 0.92 \end{array}$$

$$V_{tpz} \leftarrow CZ \text{ MOD } 3 \text{ MOD } 2 * 70 + CZ \text{ MOD } 3 \text{ DIV } 2 * 2 * 70 * 0.96 + CZ \text{ DIV } 3 * 2 * 70 * 0.96 + CZ \text{ DIV } 3 * 70 * 0.5$$

Y para los descuentos son:

$$\begin{aligned} dtoc &\leftarrow CC \text{ DIV } 3 * 3 * 15 * 0.1 & dtop &\leftarrow CP \text{ DIV } 3 * 3 * 12.5 * 0.1 \\ dtom &\leftarrow CM \text{ DIV } 5 * 5 * 10 * 0.05 & dtodvd &\leftarrow CDVD \text{ DIV } 2 * 2 * 70 * 0.08 \\ dtoz &\leftarrow CZ \text{ MOD } 3 \text{ DIV } 2 * 2 * 70 * 0.04 + CZ \text{ DIV } 3 * 2 * 70 * 0.04 + CZ \text{ DIV } 3 * 70 * 0.5 \end{aligned}$$

**Tabla 14. Comprobación de la ecuación  $V_{tpz}$  para el ejemplo 4.**

Cant.	Primer termino		Segundo termino		Tercer termino		Cuarto termino		Vtpz
	CZ MOD 3 MOD 2 * 70	=?	CZ MOD 3 DIV 2 * 2 * 70 * 0.96	=?	CZ DIV 3 * 2 * 70 * 0.96	=?	CZ DIV 3 * 2 * 70 * 0.5	=?	
0	0 MOD 3 MOD 2 * 70	0	0 MOD 3 DIV 2 * 2 * 70 * 0.96	0	0 DIV 3 * 2 * 70 * 0.96	0	0 DIV 3 * 2 * 70 * 0.5	0	0
1	1 MOD 3 MOD 2 * 70	70	1 MOD 3 DIV 2 * 2 * 70 * 0.96	0	1 DIV 3 * 2 * 70 * 0.96	0	1 DIV 3 * 2 * 70 * 0.5	0	70
2	2 MOD 3 MOD 2 * 70	0	2 MOD 3 DIV 2 * 2 * 70 * 0.96	134.4	2 DIV 3 * 2 * 70 * 0.96	0	2 DIV 3 * 2 * 70 * 0.5	0	134.4
3	3 MOD 3 MOD 2 * 70	0	3 MOD 3 DIV 2 * 2 * 70 * 0.96	0	3 DIV 3 * 2 * 70 * 0.96	134.4	3 DIV 3 * 2 * 70 * 0.5	35	169.4
4	4 MOD 3 MOD 2 * 70	70	4 MOD 3 DIV 2 * 2 * 70 * 0.96	0	4 DIV 3 * 2 * 70 * 0.96	134.4	4 DIV 3 * 2 * 70 * 0.5	35	239.4
5	5 MOD 3 MOD 2 * 70	0	5 MOD 3 DIV 2 * 2 * 70 * 0.96	134.4	5 DIV 3 * 2 * 70 * 0.96	134.4	5 DIV 3 * 2 * 70 * 0.5	35	303.8
6	6 MOD 3 MOD 2 * 70	0	6 MOD 3 DIV 2 * 2 * 70 * 0.96	0	6 DIV 3 * 2 * 70 * 0.96	268.8	6 DIV 3 * 2 * 70 * 0.5	70	338.8
7	7 MOD 3 MOD 2 * 70	70	7 MOD 3 DIV 2 * 2 * 70 * 0.96	0	7 DIV 3 * 2 * 70 * 0.96	268.8	7 DIV 3 * 2 * 70 * 0.5	70	408.8

En la Tabla 14 que el tercer y cuarto término de la ecuación para el cálculo de  $V_{tpz}$  generan un valor cuando la cantidad de zapatos es mayor o igual a 3 además, los términos primer y segundo sirven para obtener el valor a pagar tanto por una cantidad inferior a 3 así como para los mayores a este número excepto los múltiplos de 3.

Una vez obtenidas las ecuaciones que permiten resolver el problema se puede continuar con los **Elementos Involucrados**, es decir, especificar los elementos activos (usuarios) y pasivos (recursos disponibles y/o modificables) que están involucrados en el problema. Elementos Activos, el usuario que ingresará los

datos de las cantidades de cada artículo que desea comprar. Elementos pasivos, las ecuaciones anteriormente descritas.

Todas las expresiones que resuelven el ejemplo 4 están construidas con operadores algorítmicos.

Ahora se aplican las herramientas de la sección 4 para construir el algoritmo.

El algoritmo del ejemplo es:

1. Inicio.
2. Capturar la cantidad de camisas, pantalones, medias, zapatos, DVD y ropa interior que el cliente desea comprar en la tienda Wall Max.
3. Calcular el valor total de cada artículo que se compra, el valor de cada descuento obtenido y el valor total a pagar con base en lo que el cliente desea conseguir de la tienda Wall Max.
4. Visualizar el valor total de cada artículo que se compra, el valor de cada descuento obtenido y el valor total a pagar con base en lo que el cliente desea conseguir de la tienda Wall Max.
5. Fin.

Entonces el pseudocódigo queda:

Algoritmo\_tienda\_Wall\_Max

**var**

Entero: CC, CP, CM, CZ, CDVD, CRI, Vtri

Decimal: Vtpc, Vtpp, Vtpm, Vtpz, Vtpdvd, Vtri, dtoc, dtop, dtom, dtoz, dtodvd, Vt

**Inicio**

**Escribir** (“Ingrese la cantidad de camisas, pantalones, medias, zapatos, DVD y ropa interior que desea comprar en la tienda Wall Mart”)

**Leer** (CC, CP, CM, CZ, CDVD, CRI)

$Vtpc \leftarrow CC \text{ MOD } 3 * 15 + CC \text{ DIV } 3 * 3 * 15 * 0.9$

$Vtpp \leftarrow CP \text{ MOD } 3 * 12.5 + CP \text{ DIV } 3 * 3 * 12.5 * 0.9$

$Vtpm \leftarrow CM \text{ MOD } 5 * 10 + CM \text{ DIV } 5 * 5 * 10 * 0.95$

$Vtpdvd \leftarrow CDVD \text{ MOD } 2 * 70 + CDVD \text{ DIV } 2 * 2 * 70 * 0.92$

$Vtri \leftarrow CRI * 45$

$Vtpz \leftarrow CZ \text{ MOD } 3 \text{ MOD } 2 * 70 + CZ \text{ MOD } 3 \text{ DIV } 2 * 2 * 70 * 0.96 + CZ$



**DIV**  $3 * 2 * 70 * 0.96 + CZ$  **DIV**  $3 * 70 * 0.5$

**dtoc**  $\leftarrow CC$  **DIV**  $3 * 3 * 15 * 0.1$

**dtop**  $\leftarrow CP$  **DIV**  $3 * 3 * 12.5 * 0.1$

**dtom**  $\leftarrow CM$  **DIV**  $5 * 5 * 10 * 0.05$

**dtodvd**  $\leftarrow CDVD$  **DIV**  $2 * 2 * 70 * 0.08$

**dtoz**  $\leftarrow CZ$  **MOD**  $3$  **DIV**  $2 * 2 * 70 * 0.04 + CZ$  **DIV**  $3 * 2 * 70 * 0.04 + CZ$  **DIV**  $3 * 70 * 0.5$

**Vt**  $\leftarrow Vtpc + Vtpp + Vtpm + Vtdvd + Vtpz + Vtri$

**Escribir** (“El valor total a pagar por camisas, pantalones, medias, zapatos, DVD y ropa interior de su compra en la tienda Wall Mart es:”, **Vtpc**, **Vtpp**, **Vtpm**, **Vtpz**, **Vtpdvd**, **Vtri**)

**Escribir** (“El valor de descuento de las camisas, pantalones, medias, zapatos y DVD de su compra en la tienda Wall Mart es:”, **dtoc**, **dtop**, **dtom**, **dtoz**, **dtodvd**)

**Escribir** (“El valor total a pagar por su compra en la tienda Wall Mart es:”, **Vt**)

**Fin**

En la siguiente Ilustración 7 se aprecia el diagrama de flujo para este ejemplo.

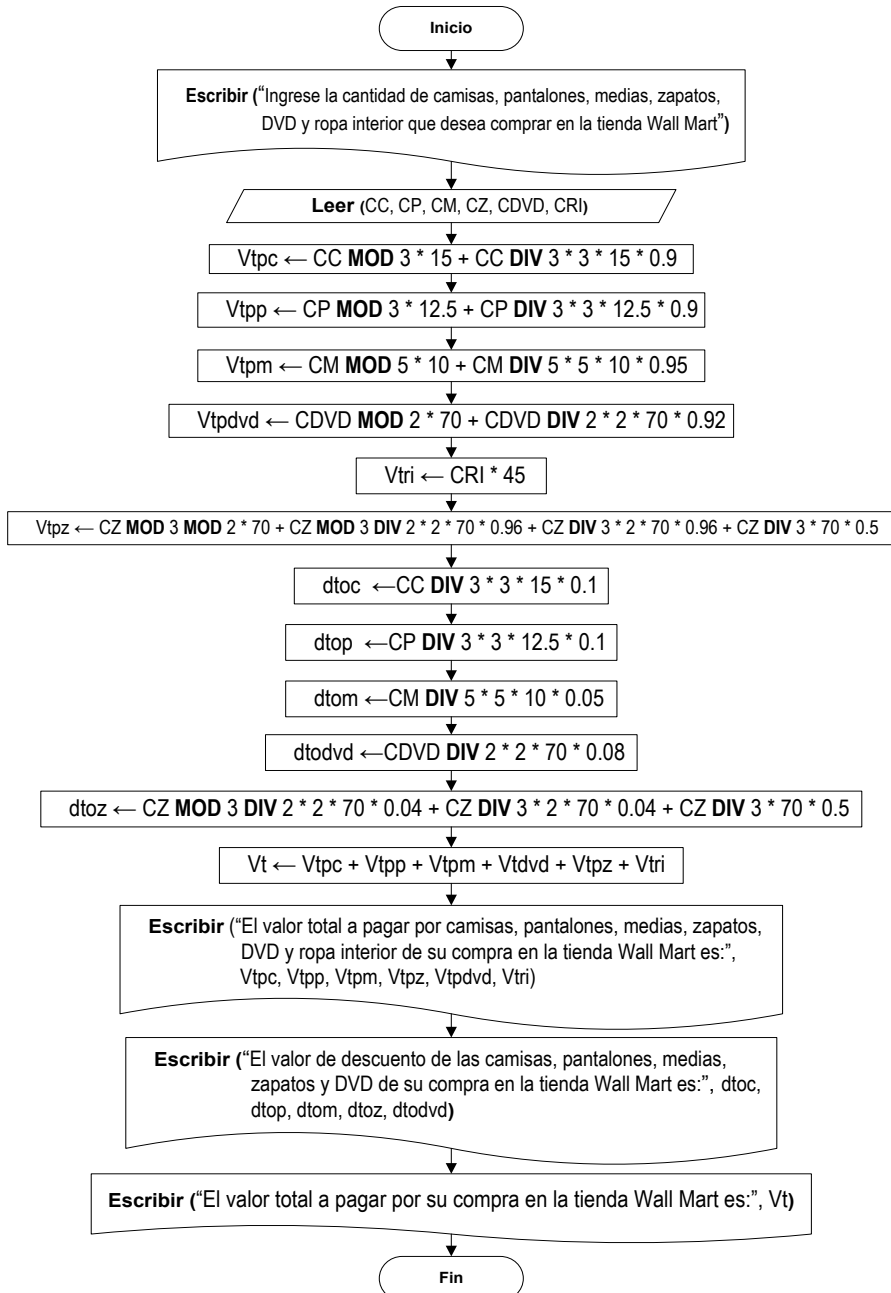


Ilustración 7. Diagrama de flujo para el ejemplo 4.