



**UNIVERSIDAD ESTATAL DE MILAGRO**  
**FACULTAD DE CIENCIAS E INGENIERIA**

**SOFTWARE C1**

**1ER**

**ASIGNATURA:**

ALGORITMO

**AUTOR:**

- NALLELY SCARLET CHICA MAYORGA
- GEOVANNA MAILYN GRACIA QUIÑONEZ
- ALEXIS JULIAN MERA ZAMORA
- JEAN PIERRE MINA QUINTERO
- BRYAN STEVEN BOHORQUEZ MANZO

**TEMA:**

EJERCICIOS

**DOCENTE:**

ING.JAIME ANDRES CAMACHO GAVILANES

**FECHA DE ENTREGA:**

20 DE JUNIO DE 2024

**Grupo:**



## Algoritmo Contador Alfabeto

```
Definir palabra, letra Como Caracter;  
  
Definir longitud_palabra, letras_alfabeto Como Entero;  
  
letras_alfabeto = 0;  
  
Escribir "Ingrese una palabra: ";  
  
Leer palabra;  
  
longitud_palabra = longitud(palabra);  
  
Para pos = 0 Hasta longitud_palabra - 1 Con Paso 1 Hacer  
    letra = subcadena(palabra, pos, pos);  
  
    Si letra >= "a" y letra <= "z" Entonces  
        letras_alfabeto = letras_alfabeto + 1;  
    FinSi  
  
Fin Para  
  
Escribir "Hay ", letras_alfabeto, " letras del alfabeto en la  
palabra ingresada.";  
  
FinAlgoritmo
```

## Algoritmo sin\_titulo

```
//1. Dado a=3 y b=7, encuentra el valor de  $y = 2 * a + b - a \bmod 3$   
  
Definir a, b Como Entero  
  
a<-3  
  
b<-7  
  
resultado<-  $2 * a + b - a \bmod 3$   
  
escribir "el resultado de y es " resultado  
  
FinAlgoritmo
```

```
1 Algoritmo sin_titulo  
2 //1. Dado a=3 y b=7, encuentra el valor de  $y = 2 * a + b - a \bmod 3$   
3 Definir a, b Como Entero  
4 a<-3  
5 b<-7  
6 resultado<-  $2 * a + b - a \bmod 3$   
7 escribir "el resultado de y es " resultado  
8 FinAlgoritmo  
9
```

PSInt - Ejecutando proceso SIN\_TITULO

```
*** Ejecución Iniciada. ***  
el resultado de y es 13  
*** Ejecución Finalizada. ***
```

## Algoritmo sin\_titulo

```
//Si a=10 y b=4, calcula el valor de  $z = a * b + 3 \bmod a + b$ .  
  
Definir a, b Como Entero  
  
a<-10  
  
b<-4  
  
resultado<-  $a * b + 3 \bmod a + b$   
  
Escribir "el resultado de z es " resultado  
  
FinAlgoritmo
```

```
1 Algoritmo sin_titulo
2 //Si a=10 y b=4, calcula el valor de  $z = a * b + 3 \bmod a + b$ .
3 Definir a, b Como Entero
4 a←10
5 b←4
6 resultado←  $a*b+3 \bmod a+b$ 
7 Escribir "el resultado de z es " resultado
8 FinAlgoritmo
9
```

PSeInt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

el resultado de z es 47

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//Con a=6 y b=2, determina el valor de  $w = a - b + 2 * a \bmod b$

Definir a, b Como Entero

a←-6

b←-2

resultado←  $a - b + 2 * a \bmod b$

Escribir "el resultado de w es " resultado

FinAlgoritmo

```
1 Algoritmo sin_titulo
2 //Con a=6 y b=2, determina el valor de  $w = a - b + 2 * a \bmod b$ 
3 Definir a, b Como Entero
4 a←6
5 b←2
6 resultado←  $a - b + 2 * a \bmod b$ 
7 Escribir "el resultado de w es " resultado
8 FinAlgoritmo
9
```

PSeInt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

el resultado de w es 4

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//4. Para a=8 y b=5, encuentra el valor de  $v = 2 * b + a \div 2 + 4 * b \bmod a$ .

Definir a, b Como Entero

a←-8

b←-5

resultado←  $2*b+a\%2+4*b \bmod a$

Escribir "el resultado de v es " resultado

FinAlgoritmo

```
1 Algoritmo sin_titulo
2 //4. Para a=8 y b=5, encuentra el valor de  $v = 2 * b + a \div 2 + 4 * b \bmod a$ .
3 Definir a, b Como Entero
4 a←8
5 b←5
6 resultado←  $2*b+a\%2+4*b \bmod a$ 
7 Escribir "el resultado de v es " resultado
8 FinAlgoritmo
```

PSeInt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

el resultado de v es 14

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//5. Si a=12 y b=9, calcula el valor de  $u = b - a + 3 * a \bmod b$ .

Definir a, b Como Entero

a←-12

b←-9

total←  $b-a+3*a \bmod b$

Escribir "el total de u es " total

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //5. Si a=12 y b=9, calcula el valor de u = b - a + 3 * a mod b.
3      Definir a, b Como Entero
4      a←12
5      b←9
6      total← b-a+3*a mod b
7      Escribir "el total de u es " total
8  FinAlgoritmo
9

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
el total de u es -3
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```
//6 ( 5 + 3 * 2 ) + 9 > 3 * 5 * 14 % 3
```

```
p←-( 5 + 3 * 2 ) + 9 > 3 * 5 * 14 % 3
```

```
Escribir "el resultado de p es " p
```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //6 ( 5 + 3 * 2 ) + 9 > 3 * 5 * 14 % 3
3      p←( 5 + 3 * 2 ) + 9 > 3 * 5 * 14 % 3
4      Escribir "el resultado de p es " p
5  FinAlgoritmo
6

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
el resultado de p es VERDADERO
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```
//7. 2 *(4 ? 10 + 8)/2* 36 *(1/2)
```

```
r←- 2 * ( 4 - 10 + 8)/2* 36 *(1/2)
```

```
Escribir "el resultado de r es " r
```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //7. 2 *(4 ? 10 + 8)/2* 36 *(1/2)
3      r←- 2 * ( 4 - 10 + 8)/2* 36 *(1/2)
4      Escribir "el resultado de r es " r
5  FinAlgoritmo
6

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
el resultado de r es 36
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```
//8. 260 / 12 + 54 % 3 - 85 % 7
```

```
t←-260/12+54%3-85%7
```

```
Escribir "el total de t es " t
```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //8. 260 / 12 + 54 % 3 - 85 % 7
3      t←260/12+54%3-85%7
4      Escribir "el total de t es " t
5  FinAlgoritmo
6

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
el total de t es 20.6666666667
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```
//9. (48 < 2 * 3) || (2 * 7 < 12)
```

```
w←-(48 < 2 * 3) | (2 * 7 < 12)
```

```
escribir "el resultado de w es " w
```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //9. (48 < 2 * 3) || (2 * 7 < 12)
3      w←(48 < 2 * 3) v (2 * 7 < 12)
4      escribir "el resultado de w es " w
5  FinAlgoritmo
6

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
el resultado de w es FALSO
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```
//10. ((8 > 2) || (932 < 23) ) && 4 == 2
```

```
Q<-((8 > 2) | (932 < 23) ) & 4 == 2
```

```
Escribir "el resultado de Q es " Q
```

FinAlgoritmo

```
1  Algoritmo sin_titulo
2  //10. ((8 > 2) || (932 < 23) ) && 4 == 2
3  Q<-((8 > 2) v (932 < 23) ) ^ 4 == 2
4  Escribir "el resultado de Q es " Q
5  FinAlgoritmo
6
```

PSelnt - Ejecutando proceso SIN\_TITULO

```
*** Ejecución Iniciada. ***
el resultado de Q es FALSO
*** Ejecución Finalizada. ***
```

//\*\*\*\*Ejercicios algoritmos secuenciales (paso a paso)\*\*\*\*\*:

Algoritmo sin\_titulo

```
// 11) Suma de dos números: Escribe un programa que tome dos números como
```

```
//entrada y muestre su suma.
```

```
Definir a, b, suma Como Entero
```

```
Escribir "ingresa el numero"
```

```
Leer a
```

```
Escribir "ingresa el numero"
```

```
Leer b
```

```
suma<-a+b
```

```
Escribir "el resultado de la suma es " suma;
```

FinAlgoritmo

```
1  Algoritmo sin_titulo
2  // 11) Suma de dos números: Escribe un programa que tome dos números como
3  //entrada y muestre su suma.
4  Definir a, b, suma Como Entero
5  Escribir "ingresa el numero"
6  Leer a
7  Escribir "ingresa el numero"
8  Leer b
9  suma<-a+b
10 Escribir "el resultado de la suma es " suma;
11 FinAlgoritmo
```

PSelnt - Ejecutando proceso SIN\_TITULO

```
*** Ejecución Iniciada. ***
ingresa el numero
> 12
ingresa el numero
> 15
el resultado de la suma es 27
*** Ejecución Finalizada. ***
```

Algoritmo sin\_titulo

```
// 12) Área de un triángulo: Pide al usuario que ingrese la base y la altura de un triángulo,
```

```
//luego calcula y muestra su área.
```

```
Definir base, altura, area Como Entero
```

```
Escribir "ingresa el valor de la base del triangulo"
```

```
Leer base
```

```
Escribir "ingresa el valor de la altura del triangulo"
```

```
Leer altura
```

```
area<-base*altura
```

```
Escribir "ingresa el valor del area del triangulo es " area;
```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  // 12) Área de un triángulo: Pide al usuario que ingrese la base y la altura de un triángulo,
3  //luego calcula y muestra su área.
4  Definir base, altura, area Como Entero
5  Escribir "ingresa el valor de la base del triángulo"
6  Leer base
7  Escribir "ingresa el valor de la altura del triángulo"
8  Leer altura
9  area=base*altura
10 Escribir "ingresa el valor del area del triángulo es " area;
11 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
ingresa el valor de la base del triángulo
> 10
ingresa el valor de la altura del triángulo
> 15
ingresa el valor del area del triángulo es 150
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

// 13) Número par o impar: Solicita al usuario

//que ingrese un número e indica si es

//par o impar.

Definir num Como Entero

Escribir "ingresa el numero"

Leer num

Si num % 2 == 0 Entonces

    Escribir "el numero es par"

SiNo

    Escribir "el numero es impar"

Fin Si

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  // 13) Número par o impar: Solicita al usuario
3  //que ingrese un número e indica si es
4  //par o impar.
5  Definir num Como Entero
6  Escribir "ingresa el numero"
7  Leer num
8  Si num % 2 == 0 Entonces
9  |   Escribir "el numero es par"
10 |
11 |   Escribir "el numero es impar"
12 |   Fin Si
13 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
ingresa el numero
> 15
el numero es impar
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

// 14) Calculadora simple: Crea una calculadora que

//realice operaciones básicas como suma, resta,

//multiplicación y división, según la elección del usuario.

Definir opc, numero1, numero2, resultado Como Entero

Escribir "elige una opcion "

Escribir "1.suma"

Escribir "2.resta"

Escribir "3.multiplicacion"

Escribir "4. división"

Escribir "ingresa una opcion"

Leer opc

Escribir "ingresa el 1er numero "

Leer numero1

Escribir "ingresa el 2do numero"

Leer numero2

Segun opc Hacer

Caso 1:

resultado<-numero1+numero2

Escribir "la suma de los valores es " resultado

Caso 2:

resultado<-numero1-numero2

Escribir "la resta de los valores es " resultado

Caso 3:

resultado<-numero1\*numero2

Escribir "la multiplicacion de los valores es " resultado

Caso 4:

resultado<-numero1/numero2

Escribir "la división de los valores es " resultado

otro:

Escribir "error"

FinSegun

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  // 14) Calculadora simple: Crea una calculadora que
3  //realice operaciones básicas como suma, resta,
4  //multiplicación y división, según la elección del usuario.
5  Definir opc, numero1, numero2, resultado Como Entero
6  Escribir "elige una opcion "
7  Escribir "1.suma"
8  Escribir "2.resta"
9  Escribir "3.multiplicacion"
10 Escribir "4. división"
11 Escribir "ingresa una opcion"
12 Leer opc
13 Escribir "ingresa el 1er numero "
14 Leer numero1
15 Escribir "ingresa el 2do numero"
16 Leer numero2
17 Segun opc Hacer
18     Caso 1:
19         resultado<=numero1+numero2
20         Escribir "la suma de los valores es " resultado
21     Caso 2:
22         resultado<=numero1-numero2
23         Escribir "la resta de los valores es " resultado
24     Caso 3:
25         resultado<=numero1*numero2
26         Escribir "la multiplicacion de los valores es " resultado
27     Caso 4:
28         resultado<=numero1/numero2
29         Escribir "la división de los valores es " resultado
30     otro:
31         Escribir "error"
32 FinSegun
33 FinAlgoritmo

```

PSelnt - Ejecutando proceso SIN\_TITULO

```

elige una opcion
1.suma
2.resta
3.multiplicacion
4. división
ingresa una opcion
> 2
ingresa el 1er numero
> 12
ingresa el 2do numero
> 40
la resta de los valores es -28
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible

Algoritmo TablaMultiplicar

//15. Tabla de multiplicar: Pide al usuario un número y muestra su tabla

//de multiplicar del 1 al 10.

Definir Num Como Entero;

Definir Resultado Como Entero;

Escribir "Ingrese un número: ";



```

Leer Numero;

// Generar y mostrar la tabla de multiplicar

Escribir "Tabla de multiplicar del ", Numero;

Para Multiplicador Desde 1 Hasta 10 Con Paso 1 Hacer

    Resultado <- Numero * Multiplicador;

    Escribir Numero, " x ", Multiplicador, " = ", Resultado;

Fin Para

```

FinAlgoritmo

```

1  Algoritmo TablaMultiplicar
2      //15. Tabla de multiplicar: Pide al usuario un número y muestra su tabla
3      //de multiplicar del 1 al 10.
4      // Definir variables
5      Definir Num Como Entero;
6      Definir Resultado Como Entero;
7
8      // Solicitar número al usuario
9      Escribir "Ingrese un número: ";
10     Leer Numero;
11
12     // Generar y mostrar la tabla de multiplicar
13     Escribir "Tabla de multiplicar del ", Numero;
14     Para Multiplicador Desde 1 Hasta 10 Con Paso 1 Hacer
15         Resultado <- Numero * Multiplicador;
16         Escribir Numero, " x ", Multiplicador, " = ", Resultado;
17     Fin Para
18
19 FinAlgoritmo

```

PSelnt - Ejecutando proceso TABLAMULTIPLICAR

```

*** Ejecución Iniciada. ***
Ingrese un número:
> 3
Tabla de multiplicar del 3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```

//16. Copiar palabra: Escribe un programa que lea dos palabras y

//concatena en otra variable las dos palabras

Definir nombre1,apellido2 , re Como Caracter

Escribir " Ingrese una nombre "

Leer nombre1

Escribir " Ingrese el apellido "

leer apellido2

re = nombre1 + apellido2

Escribir "La concatenacion de las palabras son ", re

```

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //16. Copiar palabra: Escribe un programa que lea dos palabras y
3      //concatena en otra variable las dos palabras
4      Definir nombre1,apellido2 , re Como Caracter
5      Escribir " Ingrese una nombre "
6      Leer nombre1
7      Escribir " Ingrese el apellido "
8      leer apellido2
9      re = nombre1 + apellido2
10     Escribir "La concatenacion de las palabras son ", re
11 FinAlgoritmo

```

PSelnt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese una nombre
> Josue
Ingrese el apellido
> Burgos
La concatenacion de las palabras son Josue Burgos
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```

//17. Mayor de tres números: Solicita tres números y determina cuál

//es el mayor de ellos.

Definir n1,n2,n3 Como Entero

Escribir "Ingrese el primer numero"

```



```

leer n1

Escribir "Ingrese el segundo numero"

leer n2

Escribir "Ingrese el tercer numero"

leer n3

si n1 > n2 & n1 > n3 Entonces
    Escribir "El primer numero es mayor"

SiNo
    si n2 > n1 & n2 > n3 Entonces
        Escribir "El segundo numero es mayor"
    SiNo
        Escribir "El tercer numero es mayor"

FinSi

```

FinAlgoritmo

The screenshot shows a code editor with the following algorithm:

```

1  Algoritmo sin_titulo
2  //17. Mayor de tres números: Solicita tres números y determina cuál
3  //es el mayor de ellos.
4  Definir n1,n2,n3 Como Entero
5  Escribir "Ingrese el primer numero"
6  leer n1
7  Escribir "Ingrese el segundo numero"
8  leer n2
9  Escribir "Ingrese el tercer numero"
10 leer n3
11 si n1 > n2 ^ n1 > n3 Entonces
12     Escribir "El primer numero es mayor"
13 SiNo
14     si n2 > n1 ^ n2 > n3 Entonces
15         Escribir "El segundo numero es mayor"
16     SiNo
17         Escribir "El tercer numero es mayor"
18     FinSi
19 FinSi
20 FinAlgoritmo

```

On the right, a PSeInt window titled "PSeInt - Ejecutando proceso SIN\_TITULO" shows the execution output:

```

*** Ejecución Iniciada. ***
Ingrese el primer numero
> 15
Ingrese el segundo numero
> 14
Ingrese el tercer numero
> 18
El tercer numero es mayor
*** Ejecución Finalizada. ***

```

At the bottom of the PSeInt window, there are checkboxes for "No cerrar esta ventana" and "Siempre visible".

Algoritmo sin\_titulo

```

//18. Edad mínima para votar: Pregunta la edad del usuario y
//verifica si es elegible para votar (18 años o más).

Definir edad Como Caracter

Escribir "Ingrese la edad"

leer edad

si edad >= "18" Entonces
    Escribir "Puede sufragar"

SiNo
    Escribir "No puede sufragar"

FinSi

```

FinAlgoritmo

## Algoritmo sin\_titulo

```
//18. Edad mínima para votar: Pregunta la edad del usuario y
//verifica si es elegible para votar (18 años o más).
Definir edad Como Caracter
Escribir "Ingrese la edad"
leer edad
si edad ≥ "18" Entonces
    Escribir "Puede sufragar"
SiNo
    Escribir "No puede sufragar"
FinSi
FinAlgoritmo
```

PSelnt - Ejecutando proceso SIN\_TITULO

```
*** Ejecución Iniciada. ***
Ingrese la edad
> 19
Puede sufragar
*** Ejecución Finalizada. ***
```

## Algoritmo sin\_titulo

//19. Calculadora de BMI: Crea un programa que calcule el índice de masa corporal

//(BMI) a partir del peso y la altura del usuario, y luego indique

//si está en una categoría de peso saludable.

Escribir "Ingrese su peso en kilogramos:"

Leer peso

Escribir "Ingrese su altura en metros:"

Leer altura

bmi <- peso / (altura \* altura)

// Determinar la categoría de peso

Si bmi < 18.5 Entonces

categoria <- "Bajo peso"

Sino Si bmi >= 18.5 Y bmi < 24.9 Entonces

categoria <- "Peso normal"

Sino Si bmi >= 24.9 Y bmi < 29.9 Entonces

categoria <- "Sobrepeso"

Sino

categoria <- "Obesidad"

Fin Si

fin si

fin si

Escribir "Su índice de masa corporal (BMI) es:", bmi

Escribir "Usted está en la categoría de peso:", categoria

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //19. Calculadora de BMI: Crea un programa que calcule el indice de masa corporal
3      //(BMI) a partir del peso y la altura del usuario, y luego indique
4      //si está en una categoría de peso saludable.
5      Escribir "Ingrese su peso en kilogramos:"
6      Leer peso
7      Escribir "Ingrese su altura en metros:"
8      Leer altura
9      bmi ← peso / (altura * altura)
10     // Determinar la categoría de peso
11     Si bmi < 18.5 Entonces
12         categoria ← "Bajo peso"
13     Sino Si bmi ≥ 18.5 Y bmi < 24.9 Entonces
14         categoria ← "Peso normal"
15     Sino Si bmi ≥ 24.9 Y bmi < 29.9 Entonces
16         categoria ← "Sobrepeso"
17     Sino
18         categoria ← "Obesidad"
19     Fin Si
20     fin si
21     fin si
22     Escribir "Su índice de masa corporal (BMI) es:", bmi
23     Escribir "Usted está en la categoría de peso:", categoria
24 FinAlgoritmo

```

PSelnt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

Ingrese su peso en kilogramos:

> 50

Ingrese su altura en metros:

> 130

Su índice de masa corporal (BMI) es:0.0029585799

Usted está en la categoría de peso:Bajo peso

\*\*\* Ejecución Finalizada. \*\*\*

☐ No cerrar esta ventana ☐ Siempre visible

Algoritmo sin\_titulo

//20. Número positivo, negativo o cero: Pide al usuario que ingrese

//un número y muestra si es positivo, negativo o cero.

Definir n Como Entero

Escribir "Ingrese el numero"

Leer n

Si n == 0 Entonces

Escribir "El numero es 0"

SiNo

Si n > 0 Entonces

Escribir "El numero es positivo"

SiNo

Escribir "El numero es negativo"

FinSi

FinSi

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //20. Número positivo, negativo o cero: Pide al usuario que ingrese
3      //un número y muestra si es positivo, negativo o cero.
4      Definir n Como Entero
5      Escribir "Ingrese el numero"
6      Leer n
7      Si n == 0 Entonces
8          Escribir "El numero es 0"
9      SiNo
10         Si n > 0 Entonces
11             Escribir "El numero es positivo"
12         SiNo
13             Escribir "El numero es negativo"
14         FinSi
15     FinSi
16 FinAlgoritmo

```

PSelnt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

Ingrese el numero

> 13

El numero es positivo

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo AñoBisiesto

Definir Año Como Entero;

```
Definir EsBisiesto Como Logico;

Escribir "Ingrese un año: ";

Leer Año;

Si Año <= 0 Entonces

    Escribir "Error: El año debe ser un número positivo.";

FinSi

EsBisiesto <- (Año Mod 4 == 0) Y (Año Mod 100 <> 0) O (Año Mod 400 == 0)

Si EsBisiesto Entonces

    Escribir Año, " es un año bisiesto.";

Sino

    Escribir Año, " no es un año bisiesto.";

FinSi
```

FinAlgoritmo

```
1  Algoritmo AñoBisiesto
2      Definir Año Como Entero;
3      Definir EsBisiesto Como Logico;
4      Escribir "Ingrese un año: ";
5      Leer Año;
6      Si Año ≤ 0 Entonces
7          Escribir "Error: El año debe ser un número positivo.";
8      FinSi
9      EsBisiesto ← (Año Mod 4 == 0) Y (Año Mod 100 ≠ 0) O (Año Mod 400 == 0)
10     Si EsBisiesto Entonces
11         Escribir Año, " es un año bisiesto.";
12     Sino
13         Escribir Año, " no es un año bisiesto.";
14     FinSi
15 FinAlgoritmo
```

P5eInt - Ejecutando proceso ANIOBISIESTO

```
*** Ejecución Iniciada. ***
Ingrese un año:
> 1998
1998 no es un año bisiesto.
*** Ejecución Finalizada. ***
```

Algoritmo SignoZodiacal

```
//22. Signo zodiacal: Pide al usuario que ingrese su mes y día de
//nacimiento, luego determina su signo zodiacal. Puedes usar una serie de
//declaraciones if para
//comparar las fechas ingresadas con las fechas límite de cada signo
//zodiacal.

Definir Mes, Dia Como Entero;

Definir SignodeZodiaco Como Caracter;

Escribir "Ingrese su mes de nacimiento (en número): ";

Leer Mes;

Si Mes < 1 O Mes > 12 Entonces

    Escribir "Error: El mes debe estar entre 1 y 12.";

FinSi

Escribir "Ingrese su día de nacimiento: ";

Leer Dia;

Si Mes == 1 Entonces

    Si Dia <= 19 Entonces
```

SignodeZodiaco <- "Capricornio";

Sino

SignodeZodiaco <- "Acuario";

FinSi

Sino

Si Mes == 2 Entonces

Si Dia <= 18 Entonces

SignodeZodiaco <- "Acuario";

Sino

SignodeZodiaco <- "Piscis";

FinSi

Sino

Si Mes == 3 Entonces

Si Dia <= 19 Entonces

SignodeZodiaco <- "Piscis";

Sino

SignodeZodiaco <- "Aries";

FinSi

Sino

Si Mes == 4 Entonces

Si Dia <= 19 Entonces

SignodeZodiaco <- "Aries";

Sino

SignodeZodiaco <- "Tauro";

FinSi

Sino

Si Mes == 5 Entonces

Si Dia <= 20 Entonces

SignodeZodiaco <- "Tauro";

Sino

SignodeZodiaco <- "Géminis";

FinSi

Sino

Si Mes == 6 Entonces

Si Dia <= 20 Entonces

SignodeZodiaco <- "Géminis";

Sino

SignodeZodiaco <- "Cáncer";

FinSi

Sino

Si Mes == 7 Entonces

Si Dia <= 22 Entonces

SignodeZodiaco <- "Cáncer";

Sino

SignodeZodiaco <- "Leo";

FinSi

Sino

Si Mes == 8 Entonces

Si Dia <= 22 Entonces

SignodeZodiaco <- "Leo";

Sino

SignodeZodiaco <- "Virgo";

FinSi

Sino

Si Mes == 9 Entonces

Si Dia <= 22 Entonces

SignodeZodiaco <- "Virgo";

Sino

SignodeZodiaco <- "Libra";

FinSi

Sino

Si Mes == 10 Entonces

Si Dia <= 22 Entonces

SignodeZodiaco <- "Libra";

Sino

SignodeZodiaco <- "Escorpio";

FinSi

Sino

Si Mes == 11 Entonces

Si Dia <= 21 Entonces

SignodeZodiaco <- "Escorpio";

Sino

SignodeZodiaco <- "Sagitario";

FinSi

Sino

Si Mes == 12 Entonces

Si Dia <= 21 Entonces

```
SignodeZodiaco <- "Sagitario";
```

```
Sino
```

```
SignodeZodiaco <- "Capricornio";
```

```
FinSi
```

```
FinSi
```

```
Finsi
```

```
Finsi
```

```
FinSi
```

```
FinSi
```

```
FinSi
```

```
FinSi
```

```
FinSi
```

```
FinSi
```

```
FinSi
```

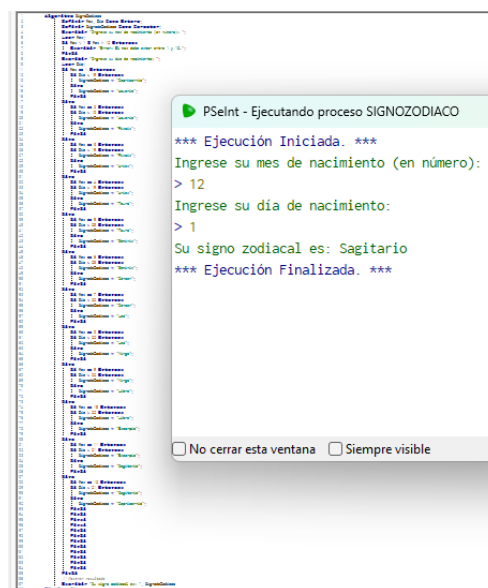
```
FinSi
```

```
FinSi
```

```
// Mostrar resultado
```

```
Escribir "Su signo zodiacal es: ", SignodeZodiaco
```

FinAlgoritmo



Algoritmo DiaQuincena

```
//23. Día del mes con respecto a la segunda quincena: Solicita al usuario que ingrese
```

```
//un número de día del mes (por ejemplo, del 1 al 31) y verifica si ese día pertenece a la primera quincena
```

```
(días 1-15) o a la segunda quincena (días 16-31).
```

```
Definir Dia Como Entero;
```

```
Definir Quincena Como Caracter;
```

```
// Solicitar día del mes
```



```

Escribir "Ingrese un día del mes (1 al 31): ";

Leer Dia;

// Validar entrada de día (entre 1 y 31)

Si Dia < 1 O Dia > 31 Entonces

Escribir "Error: El día debe estar entre 1 y 31.";

FinSi

// Determinar quincena

Si Dia <= 15 Entonces

    Quincena <- "Primera quincena";

Sino

    Quincena <- "Segunda quincena";

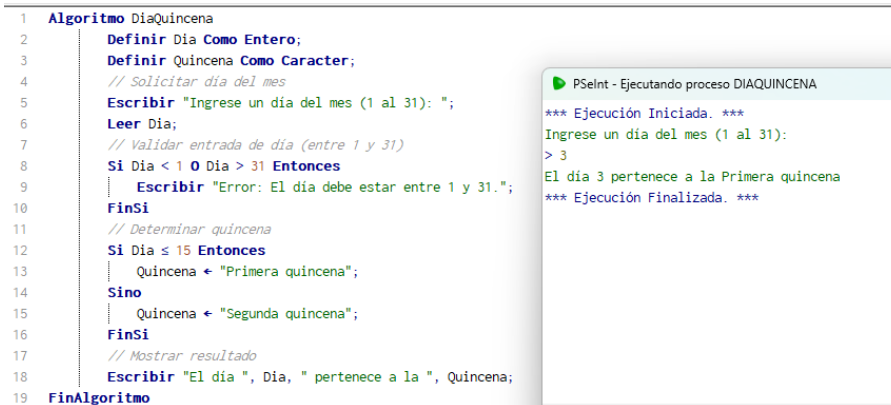
FinSi

// Mostrar resultado

Escribir "El día ", Dia, " pertenece a la ", Quincena;

```

FinAlgoritmo



```

1  Algoritmo DiaQuincena
2      Definir Dia Como Entero;
3      Definir Quincena Como Caracter;
4      // Solicitar día del mes
5      Escribir "Ingrese un día del mes (1 al 31): ";
6      Leer Dia;
7      // Validar entrada de día (entre 1 y 31)
8      Si Dia < 1 O Dia > 31 Entonces
9          Escribir "Error: El día debe estar entre 1 y 31.";
10     FinSi
11     // Determinar quincena
12     Si Dia ≤ 15 Entonces
13         Quincena ← "Primera quincena";
14     Sino
15         Quincena ← "Segunda quincena";
16     FinSi
17     // Mostrar resultado
18     Escribir "El día ", Dia, " pertenece a la ", Quincena;
19 FinAlgoritmo

```

PSelnt - Ejecutando proceso DIAQUINCENA

```

*** Ejecución Iniciada. ***
Ingrese un día del mes (1 al 31):
> 3
El día 3 pertenece a la Primera quincena
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

```

//24. Día de la semana: Pide al usuario que ingrese un número del 1 al 7, donde 1
//representa el domingo, 2 el lunes, 3 el martes, y así sucesivamente.

//Luego, utiliza una estructura switch para mostrar el nombre del día de la
//semana

//correspondiente al número ingresado.

```

```

Definir n Como Entero

Escribir "elige un numero "

leer n

Segun n hacer

    caso 1:

        Escribir "Domingo"

    caso 2:

        Escribir "Lunes"

```

```

caso 3 :
    Escribir "Martes"

caso 4 :
    Escribir "Miercoles"

caso 5:
    Escribir "Jueve"

caso 6 :
    Escribir "Viernes"

caso 7 :
    Escribir "Sabado"

otro:
    Escribir "Opcion no validad"

```

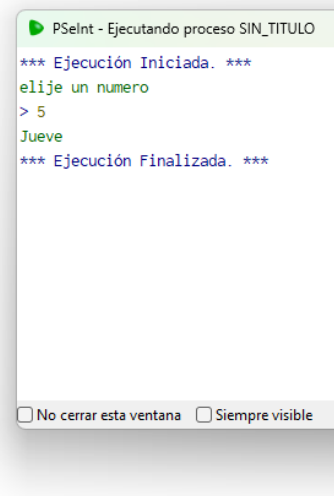
FinSegun

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //24. Día de la semana: Pide al usuario que ingrese un número del 1 al 7, donde 1
3      //representa el domingo, 2 el lunes, 3 el martes, y así sucesivamente.
4      //Luego, utiliza una estructura switch para mostrar el nombre del día de la
5      //semana
6      //correspondiente al número ingresado.
7      Definir n Como Entero
8      Escribir "elige un numero "
9      leer n
10     Segun n hacer
11         caso 1:
12             Escribir "Domingo"
13         caso 2:
14             Escribir "Lunes"
15         caso 3 :
16             Escribir "Martes"
17         caso 4 :
18             Escribir "Miercoles"
19         caso 5:
20             Escribir "Jueve"
21         caso 6 :
22             Escribir "Viernes"
23         caso 7 :
24             Escribir "Sabado"
25         otro:
26             Escribir "Opcion no validad"
27     FinSegun
28 FinAlgoritmo

```



Algoritmo sin\_titulo

//25. Frases iguales: Escribir un programa que ingrese dos frases e

//indique si son iguales

```

Definir palabra1,palabra2 ,res Como Caracter

Escribir "Ingresa la primera frase"

leer palabra1

Escribir "Ingresa la segunda frase"

leer palabra2

si Longitud(palabra1) == Longitud(palabra2) Entonces

```

Escribir "Si son iguales"

SiNo

Escribir "No son iguales"

FinSi

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //25. Frases iguales: Escribir un programa que ingrese dos frases e
3  //indique si son iguales
4      Definir palabra1,palabra2 ,res Como Caracter
5      Escribir "Ingrese la primera frase"
6      leer palabra1
7      Escribir "Ingrese la segunda frase"
8      leer palabra2
9      si Longitud(palabra1) == Longitud(palabra2) Entonces
10         Escribir "Si son iguales"
11     SiNo
12         Escribir "No son iguales"
13     FinSi
14 FinAlgoritmo
15

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese la primera frase
> hi
Ingrese la segunda frase
> hi
Si son iguales
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//26. Calculadora de precio con descuento: Crea un programa que permita a

//un usuario ingresar

//el precio de un artículo y un porcentaje de descuento. El

//programa debe calcular y mostrar el precio final después del descuento.

Definir Valor, descuento, valor\_total Como Entero

Escribir "Ingrese el valor"

leer Valor

Escribir "Ingrese el descuento"

leer descuento

valor\_total<- Valor- descuento

Escribir "El valor total a pagar es ", valor\_total

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //26. Calculadora de precio con descuento: Crea un programa que permita a
3  //un usuario ingresar
4  //el precio de un artículo y un porcentaje de descuento. El
5  //programa debe calcular y mostrar el precio final después del descuento.
6  Definir Valor, descuento, valor_total Como Entero
7  Escribir "Ingrese el valor"
8  leer Valor
9  Escribir "Ingrese el descuento"
10 leer descuento
11 valor_total<- Valor- descuento
12 Escribir "El valor total a pagar es ", valor_total
13 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese el valor
> 100
Ingrese el descuento
> 12
El valor total a pagar es 88
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//27. Calculadora de factura con impuestos: Solicita al usuario que

//ingrese el total de una factura y el porcentaje de impuestos aplicado.

//Luego, calcula y muestra

//el monto total a pagar, incluyendo los impuestos.

```
Definir fac, imput ,monto_t Como Entero

Escribir "Ingrese la factura"

leer fac

Escribir "Ingrese el impuesto"

leer imput

monto_t <- fac + imput

Escribir "El monto total a pagar es ", monto_t
```

FinAlgoritmo

```
1 Algoritmo sin_titulo
2 //27. Calculadora de factura con impuestos: Solicita al usuario que
3 //ingrese el total de una factura y el porcentaje de impuestos aplicado.
4 //Luego, calcula y muestra
5 //el monto total a pagar, incluyendo los impuestos.
6 Definir fac, imput ,monto_t Como Entero
7 Escribir "Ingrese la factura"
8 leer fac
9 Escribir "Ingrese el impuesto"
10 leer imput
11 monto_t <- fac + imput
12 Escribir "El monto total a pagar es ", monto_t
13 FinAlgoritmo
```

PSeInt - Ejecutando proceso SIN\_TITULO

```
*** Ejecución Iniciada. ***
Ingrese la factura
> 100
Ingrese el impuesto
> 15
El monto total a pagar es 115
*** Ejecución Finalizada. ***
```

Algoritmo CalculadoraSueldoAumento

// Definir variables

Definir SueldoActua, PorcentajeAumento, NuevoSueldol Como Real;

// Solicitar salario actual

Escribir "Ingrese su salario actual: ";

Leer SueldoActual;

// Validar entrada de salario (mayor que 0)

Si SueldoActual <= 0 Entonces

Escribir "Error: El salario debe ser un valor mayor que 0.";

FinSi

// Solicitar porcentaje de aumento

Escribir "Ingrese el porcentaje de aumento (en número): ";

Leer PorcentajeAumento;

// Validar entrada de porcentaje (entre 0 y 100)

Si PorcentajeAumento < 0 O PorcentajeAumento > 100 Entonces

Escribir "Error: El porcentaje de aumento debe estar entre 0 y 100.";

FinSi

// Calcular nuevo salario

NuevoSueldo <- SueldoActual + (SueldoActual \* (PorcentajeAumento / 100));

// Mostrar resultado

Escribir "Su nuevo salario después del aumento es de: ", NuevoSueldo;

FinAlgoritmo

```

1  Algoritmo CalculadoraSueldoAumento
2  // Definir variables
3  + Definir SueldoActual, PorcentajeAumento, NuevoSueldo Como Real;
4  // Solicitar salario actual
5  Escribir "Ingrese su salario actual: ";
6  Leer SueldoActual;
7  // Validar entrada de salario (mayor que 0)
8  Si SueldoActual <= 0 Entonces
9  | Escribir "Error: El salario debe ser un valor mayor que 0.";
10 | FinSi
11 // Solicitar porcentaje de aumento
12 Escribir "Ingrese el porcentaje de aumento (en número): ";
13 Leer PorcentajeAumento;
14 // Validar entrada de porcentaje (entre 0 y 100)
15 Si PorcentajeAumento < 0 O PorcentajeAumento > 100 Entonces
16 | Escribir "Error: El porcentaje de aumento debe estar entre 0 y 100.";
17 | FinSi
18 // Calcular nuevo salario
19 NuevoSueldo ← SueldoActual + (SueldoActual * (PorcentajeAumento / 100));
20 // Mostrar resultado
21 Escribir "Su nuevo salario después del aumento es de: ", NuevoSueldo;
22 FinAlgoritmo

```

PSelnt - Ejecutando proceso CALCULADORASUELDOAUMENTO

```

*** Ejecución Iniciada. ***
Ingrese su salario actual:
> 1500
Ingrese el porcentaje de aumento (en número):
> 12
Su nuevo salario después del aumento es de: 1680
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible

## Algoritmo sin\_titulo

//29. Calculadora de compra con múltiples artículos: Permite al usuario

//ingresar el precio y la cantidad de varios artículos que está comprando.

//Calcula el total de la compra y aplica un descuento del 10% si el total

//es mayor a cierta cantidad (por ejemplo, \$100).

Definir valor, cantidad, subtotal, total, descuento Como Real

Definir cantidaddeArticulos, i Como Entero

total <- 0

Escribir "Ingrese la cantidad de artículos que está comprando:"

Leer cantidaddeArticulos

Para i <- 1 Hasta cantidaddeArticulos Hacer

    Escribir "Artículo ", i

    Escribir "Ingrese el precio del artículo:"

    Leer valor

    Escribir "Ingrese la cantidad del artículo:"

    Leer cantidad

    subtotal <- valor \* cantidad

    total <- total + subtotal

Fin Para

Si total > 100 Entonces

    descuento <- total \* 0.10

    total <- total - descuento

    Escribir "Se aplicó un descuento del 10%."

Fin Si

Escribir "El total de la compra es: \$", total

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //29. Calculadora de compra con multiples articulos: Permite al usuario
3  //ingresar el precio y la cantidad de varios articulos que está comprando.
4  //Calcula el total de la compra y aplica un descuento del 10% si el total
5  //es mayor a cierta cantidad (por ejemplo, $100).
6  Definir valor, cantidad, subtotal, total, descuento Como Real
7  Definir cantidaddeArticulos, i Como Entero
8  total ← 0
9  Escribir "Ingrese la cantidad de articulos que está comprando:"
10 Leer cantidaddeArticulos
11 Para i ← 1 Hasta cantidaddeArticulos Hacer
12     Escribir "Artículo ", i
13     Escribir "Ingrese el precio del artículo:"
14     Leer valor
15     Escribir "Ingrese la cantidad del artículo:"
16     Leer cantidad
17     subtotal ← valor * cantidad
18     total ← total + subtotal
19 Fin Para
20 Si total > 100 Entonces
21     descuento ← total * 0.10
22     total ← total - descuento
23     Escribir "Se aplicó un descuento del 10%."
24 Fin Si
25 Escribir "El total de la compra es: $", total
26 FinAlgoritmo
  
```

Algoritmo sin\_titulo

//30. Calculadora de impuestos sobre el salario: Solicita al usuario que

//ingrese su salario anual y calcula el impuesto sobre la renta según las

//siguientes tasas: hasta De \$10,001 a \$20,000: 1 \$10,000: 5% 0% . Más de

//\$20,000: 15%

Definir salarioAnual, impuestoRenta Como Real

Escribir "Ingrese su salario anual:"

Leer salarioAnual

Si salarioAnual <= 10000 Entonces

    impuestoRenta = 0

Sino

    Si salarioAnual <= 20000 Entonces

        impuestoRenta = (salarioAnual - 10000) \* 0.05

    Sino

        impuestoRenta = (20000 - 10000) \* 0.05 + (salarioAnual - 20000) \* 0.15

    FinSi

FinSi

Escribir "El impuesto sobre la renta es: ", impuestoRenta

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //30. Calculadora de impuestos sobre el salario: Solicita al usuario que
3      //ingrese su salario anual y calcula el impuesto sobre la renta según las
4      //siguientes tasas: hasta De $10,001 a $20,000: 1 $10,000: 5% 0% . Más de
5      //$20,000: 15%
6      Definir salarioAnual, impuestoRenta Como Real
7      Escribir "Ingrese su salario anual:"
8      Leer salarioAnual
9      Si salarioAnual ≤ 10000 Entonces
10         impuestoRenta = 0
11     Sino
12         Si salarioAnual ≤ 20000 Entonces
13             impuestoRenta = (salarioAnual - 10000) * 0.05
14         Sino
15             impuestoRenta = (20000 - 10000) * 0.05 + (salarioAnual - 20000) * 0.15
16         FinSi
17     FinSi
18     Escribir "El impuesto sobre la renta es: ", impuestoRenta
19 FinAlgoritmo

```

PSInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese su salario anual:
> 20500
El impuesto sobre la renta es: 575
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//31. Descuento por antigüedad en la empresa: Pregunta al usuario cuántos

//años ha estado trabajando en una empresa y calcula su bono de antigüedad.

//Si ha trabajado más de 5 años, otorga un bono del 5% sobre su salario.

Definir añosdetrabajo , sueldo\_final ,bono , sueldo Como Entero

Escribir " Ingrese los años de trabajo"

leer añosdetrabajo

Escribir "Ingrese su sueldo"

leer sueldo

si añosdetrabajo > 5 Entonces

    bono <- sueldo \* 0.05

    sueldo\_final <- sueldo + bono

    Escribir " Sueldo final mas bono ", sueldo\_final

sino

    Escribir " Su sueldo es ", sueldo

FinSi

FinAlgoritmo

```

1  Algoritmo sin_titulo
2      //31. Descuento por antigüedad en la empresa: Pregunta al usuario cuántos
3      //años ha estado trabajando en una empresa y calcula su bono de antigüedad.
4      //Si ha trabajado más de 5 años, otorga un bono del 5% sobre su salario.
5      Definir añosdetrabajo , sueldo_final ,bono , sueldo Como Entero
6      Escribir " Ingrese los años de trabajo"
7      leer añosdetrabajo
8      Escribir "Ingrese su sueldo"
9      leer sueldo
10     si añosdetrabajo > 5 Entonces
11         bono <- sueldo * 0.05
12         sueldo_final <- sueldo + bono
13         Escribir " Sueldo final mas bono ", sueldo_final
14     sino
15         Escribir " Su sueldo es ", sueldo
16     FinSi
17 FinAlgoritmo

```

PSInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese los años de trabajo
> 20
Ingrese su sueldo
> 5000
Sueldo final mas bono 5250
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//32. Calculadora de envío con tarifas diferentes: Crea un programa que permita al usuario ingresar

//la distancia de envío y calcule el costo del envío. Si la distancia es inferior a 50 km,

//el costo es de \$10. Si la distancia es de 50 km o más, el costo es de \$20.

Definir km Como Entero



Escribir "Ingrese la distancia"

leer km

si km < 50 Entonces

Escribir " El costo del envio es \$10 dolares"

sino

si km >= 50 Entonces

Escribir " El costo del envio es de \$ 20 dolares"

FinSi

FinSi

FinAlgoritmo

```

1 Algoritmo sin_titulo
2 //32. Calculadora de envio con tarifas diferentes: Crea un programa que permita al usuario ingresar
3 //la distancia de envio y calcule el costo del envio. Si la distancia es inferior a 50 km,
4 //el costo es de $10. Si la distancia es de 50 km o más, el costo es de $20.
5 Definir km Como Entero
6 Escribir "Ingrese la distancia"
7 leer km
8 si km < 50 Entonces
9     Escribir " El costo del envio es $10 dolares"
10 sino
11     si km >= 50 Entonces
12         Escribir " El costo del envio es de $ 20 dolares"
13     FinSi
14 FinSi
15 FinAlgoritmo
  
```

PSeInt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

Ingrese la distancia

> 60

El costo del envio es de \$ 20 dolares

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//33. Calculadora de descuento por lealtad del cliente: Pide al usuario

//que ingrese el total de sus compras mensuales durante un año. Si el total

//es superior a \$500, aplica un descuento del 10% en la próxima compra.

Definir total\_compras , por\_descuento Como Entero

Escribir "Ingresa de compras mensual "

leer total\_compras

si total\_compras > 500 Entonces

por\_descuento <- total\_compras \* 0.10

Escribir " Querrido cliente usted a ganado un descuento del 10% en su proxima compra

la cual seria de ", por\_descuento

SiNo

Escribir "Estimado cliente no ha obtenido ningun descuento"

fin si

FinAlgoritmo

```

1 Algoritmo sin_titulo
2 //33. Calculadora de descuento por lealtad del cliente: Pide al usuario
3 //que ingrese el total de sus compras mensuales durante un año. Si el total
4 //es superior a $500, aplica un descuento del 10% en la próxima compra.
5 Definir total_compras , por_descuento Como Entero
6 Escribir "Ingresa de compras mensual "
7 leer total_compras
8 si total_compras > 500 Entonces
9     por_descuento <- total_compras * 0.10
10     Escribir " Querrido cliente usted a ganado un descuento del 10% en su proxima compra la cual seria de ", por_descuento
11 SiNo
12     Escribir "Estimado cliente no ha obtenido ningun descuento"
13 fin si
14 FinAlgoritmo
15
  
```

PSeInt - Ejecutando proceso SIN\_TITULO

\*\*\* Ejecución Iniciada. \*\*\*

Ingresa de compras mensual

> 200

Estimado cliente no ha obtenido ningun descuento

\*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//34. Calculadora de descuento por volumen de compra: Permite al usuario

//ingresar la cantidad de unidades de un producto que va a comprar y el

//precio unitario. Aplica descuentos por volumen de compra según las

//siguientes reglas:10-50 unidades: 5% de descuento

//51-100 unidades: 10% de descuento

// Más de 100 unidades: 15% de descuento

Definir precio\_unitario, cantidad\_producto, total\_pago, descuentos Como Real

total <- 0

Escribir "Artículo "

Escribir "Ingrese el precio del artículo:"

Leer precio\_unitario

Escribir "Ingrese la cantidad del artículo:"

Leer cantidad\_producto

si cantidad\_producto >= 10 y cantidad\_producto <= 50 Entonces

total\_pago <- precio\_unitario \* cantidad\_producto

descuentos <- total\_pago \* 0.05

total\_pago <- total\_pago - descuentos

Escribir " total a pagar con su descuento es ", total\_pago

FinSi

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //34. Calculadora de descuento por volumen de compra: Permite al usuario
3  //ingresar la cantidad de unidades de un producto que va a comprar y el
4  //precio unitario. Aplica descuentos por volumen de compra según las
5  //siguientes reglas:10-50 unidades: 5% de descuento
6  //51-100 unidades: 10% de descuento
7  // Más de 100 unidades: 15% de descuento
8  Definir precio_unitario, cantidad_producto, total_pago, descuentos Como Real
9  total <- 0
10 Escribir "Artículo "
11 Escribir "Ingrese el precio del artículo:"
12 Leer precio_unitario
13 Escribir "Ingrese la cantidad del artículo:"
14 Leer cantidad_producto
15 si cantidad_producto >= 10 y cantidad_producto <= 50 Entonces
16     total_pago <- precio_unitario * cantidad_producto
17     descuentos <- total_pago * 0.05
18     total_pago <- total_pago - descuentos
19     Escribir " total a pagar con su descuento es ", total_pago
20 FinSi
21 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Artículo
Ingrese el precio del artículo:
> 12
Ingrese la cantidad del artículo:
> 15
total a pagar con su descuento es 171
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//35.Calculadora de costo de servicio: Pregunta al usuario cuántas horas

//de servicio necesita y calcula el costo total. Si las horas son más de

//10, aplica un descuento del 20%.

Definir horadeservicio, descuento, costo\_total, costo Como Entero

tarifa = 10

Escribir " Cuantas hora de servicio necesita"

leer horadeservicio

Si horadeservicio > 10 Entonces

costo <- horadeservicio \* tarifa

descuento <- costo \* 0.20

```
costo_total <- costo - descuento

Escribir "El costo total con descuento " , costo_total
```

SiNo

```
costo <- horadeservicio * tarifa

Escribir " El costo es " , costo
```

FinSi

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //35. Calculadora de costo de servicio: Pregunta al usuario cuántas horas
3  //de servicio necesita y calcula el costo total. Si las horas son más de
4  //10, aplica un descuento del 20%.
5  Definir horadeservicio, descuento, costo_total, costo Como Entero
6  tarifa = 10
7  Escribir " Cuantas hora de servicio necesita"
8  leer horadeservicio
9  Si horadeservicio > 10 Entonces
10     costo <- horadeservicio * tarifa
11     descuento <- costo * 0.20
12     costo_total <- costo - descuento
13     Escribir "El costo total con descuento " , costo_total
14 SiNo
15     costo <- horadeservicio * tarifa
16     Escribir " El costo es " , costo
17 FinSi
18 FinAlgoritmo
  
```

PSInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Cuantas hora de servicio necesita
> 12
El costo total con descuento 96
*** Ejecución Finalizada. ***
  
```

Algoritmo Ciclos

```
//****Ejecucion del ciclo for(para)***
```

Algoritmo SumaParesFor

```
//36. Suma de números pares: Utiliza un bucle for para calcular la suma
```

```
//de los números pares del 1 al 50.
```

```

// Definir variables

Definir Contador Como Entero;

Definir SumaPares Como Entero;

// Inicializar variables

Contador <- 1;

SumaPares <- 0;

// Bucle For para iterar del 1 al 50

Para Contador Desde 1 Hasta 50 Con Paso 1 Hacer

// Verificar si el número actual es par

Si Contador Mod 2 == 0 Entonces

    // Sumar el número par a la variable SumaPares

    SumaPares <- SumaPares + Contador;

FinSi

Fin Para

// Mostrar resultado

Escribir "La suma de los números pares del 1 al 50 es: " , SumaPares;
  
```

FinAlgoritmo

```

1  Algoritmo SumaParesFor
2  //36. Suma de números pares: Utiliza un bucle for para calcular la suma
3  //de los números pares del 1 al 50.
4  // Definir variables
5  Definir Contador Como Entero;
6  Definir SumaPares Como Entero;
7  // Inicializar variables
8  Contador ← 1;
9  SumaPares ← 0;
10 // Bucle For para iterar del 1 al 50
11 Para Contador Desde 1 Hasta 50 Con Paso 1 Hacer
12     // Verificar si el número actual es par
13     Si Contador Mod 2 == 0 Entonces
14         // Sumar el número par a la variable SumaPares
15         SumaPares ← SumaPares + Contador;
16     FinSi
17 Fin Para
18 // Mostrar resultado
19 Escribir "La suma de los números pares del 1 al 50 es: ", SumaPares;
20 FinAlgoritmo

```

PSInt - Ejecutando proceso SUMAPARESFOR

```

*** Ejecución Iniciada. ***
La suma de los números pares del 1 al 50 es: 650
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible

Algoritmo sin\_titulo

//37. Tabla de multiplicar: Utiliza un bucle for para imprimir la tabla

//de multiplicar de un número ingresado por el usuario del 1 al 12

Definir m Como Entero;

Escribir "Para Obtener las Tablas De Multiplicar";

Escribir "ingrese un numero: ";

Leer m;

//Estructura Repetitiva

Para i<-1 Hasta 12 Hacer

Mostrar m,"\*",i,"=","i\*m;

Fin Para

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //37. Tabla de multiplicar: Utiliza un bucle for para imprimir la tabla
3  //de multiplicar de un número ingresado por el usuario del 1 al 12
4  Definir m Como Entero;
5  Escribir "Para Obtener las Tablas De Multiplicar";
6  Escribir "ingrese un numero: ";
7  Leer m;
8  //Estructura Repetitiva
9  Para i<-1 Hasta 12 Hacer
10     Mostrar m,"*",i,"=","i*m;
11 Fin Para
12 FinAlgoritmo
13

```

PSInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Para Obtener las Tablas De Multiplicar
ingrese un numero:
> 4
4*1=4
4*2=8
4*3=12
4*4=16
4*5=20
4*6=24
4*7=28
4*8=32
4*9=36
4*10=40
4*11=44
4*12=48
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//38. Contador de vocales: Utiliza un bucle while para contar el número

//de vocales en una palabra ingresada por el usuario.

Definir text, l Como Caracter

Definir i Como Entero

c <- 0

c1 <- 0

c2 <- 0

c3 <- 0

c4 <- 0

```
// Instrucción de entrada

Escribir "Ingrese un texto"

Leer text

i <- 1

// Estructura Repetitiva

Mientras i <= Longitud(text) Hacer

    l <- Subcadena(text, i, i)

    Si l = "a" Entonces

        c <- c + 1

    Sino

        Si l = "e" Entonces

            c1 <- c1 + 1

        Sino

            Si l = "i" Entonces

                c2 <- c2 + 1

            SiNo

                Si l = "o" Entonces

                    c3 <- c3 + 1

                SiNo

                    Si l = "u" Entonces

                        c4 <- c4 + 1

                    FinSi

                FinSi

            FinSi

        FinSi

    FinSi

    i <- i + 1

FinMientras

Mostrar "La cantidad de a =", c

Mostrar "La cantidad de e =", c1

Mostrar "La cantidad de i =", c2

Mostrar "La cantidad de o =", c3

Mostrar "La cantidad de u =", c4

FinAlgoritmo
```

```

1  Algoritmo [Sin_Titulo]
2  //30. Contador de vocales: utiliza un bucle while para contar el número
3  //de vocales en una palabra ingresada por el usuario.
4
5  Definir: text, i Como Caracter
6
7  Definir i Como Entero
8
9  c ← 0
10 c1 ← 0
11 c2 ← 0
12 c3 ← 0
13 c4 ← 0
14
15 //Estructura de entrada
16 Escribir "Ingrese un texto"
17 Leer text
18 i ← 1
19
20 //Estructura Repetitiva
21 Mientras i ≤ Longitud(text) Hacer
22     i ← Subcadena(text, i, i)
23     Si i = "a" Entonces
24         c ← c + 1
25     Sino
26         Si i = "e" Entones
27             c1 ← c1 + 1
28         Sino
29             Si i = "i" Entones
30                 c2 ← c2 + 1
31             Sino
32                 Si i = "o" Entones
33                     c3 ← c3 + 1
34                 Sino
35                     Si i = "u" Entones
36                         c4 ← c4 + 1
37                     FinSi
38                 FinSi
39             FinSi
40         FinSi
41     FinSi
42     i ← i + 1
43
44 FinMientras
45
46 Muestra "La cantidad de a = ", c
47 Muestra "La cantidad de e = ", c1
48 Muestra "La cantidad de i = ", c2
49 Muestra "La cantidad de o = ", c3
50 Muestra "La cantidad de u = ", c4
51
52 FinAlgoritmo

```

# Algoritmo sin\_titulo

//39. Contador de digitos: Utiliza un bucle for para contar el numero de

```
//dígitos en una palabra ingresada por el usuario.
```

Definir frase, frasemin Como Caracter;

Definir la, vocales, consonan, digitos, pos,vacio Como Entero;

```
vocales = 0;
```

```
consonan = 0;
```

```
digitos = 0;
```

vacio = 0;

Escribir "Ingrese la frase ";

leer frase;

```
la = longitud(frase);
```

```
frasemin = Minusculas(frase);
```

Para pos = 0 Hasta la-1 Con Paso 1 Hacer

Si subcadena(frasemin,pos,pos) >= "0" y subcadena(frasemin,pos,pos) <= "9" Entonces

```
digitos = digitos + 1;
```

SiNo

Si Subcadena(frasemin,pos,pos) = " " o subcadena(frasemin,pos,pos) = "," o

subcadena(frasemin,pos,pos) = "." o subcadena(frasemin,pos,pos) = ";" o subcadena(frasemin,pos,pos)= ":"

## Entonces

```
vacio = vacio + 1;
```

FinSi

FinSi

Fin Para

Escribir " Hay " , digitos , " digitos en la frase":

FinAlgoritmo



```
P$eInt - Ejecutando proceso SIMV_TITULO

*** Ejecución Iniciada. ***
Ingrese la frase
> 2. hola
Hay 1 digitos en la frase
*** Ejecución Finalizada. ***
```

## Proceso AdivinaElNumero

//40. Adivina el número: Genera un número aleatorio y pide al usuario que

```
//adivine el número. Utiliza un bucle while para repetir la solicitud hasta
```

```
//que adivine correctamente.
```

Definir numero\_secreto, intento, intentos\_restantes Como Entero

Definir adivinanza Como Entero

```
// Generar número aleatorio entre 1 y 100
```

```
numero_secreto = Azar(100)
```

```
// Inicializar intentos restantes
```

```
intentos_restantes = 3
```

```
// Bucle para los intentos
```

Mientras intentos\_restantes > 0 Hacer

```
// Pedir al usuario que adivine
```

Escribir "Adivina el número entre 1 y 100: "

Leer adivinanza

```
// Verificar si la adivinanza es correcta
```

Si adivinanza = numero\_secreto Entonces

Escribir "¡Correcto! Has adivinado el número."

```
// Salir del bucle si la adivinanza es correcta
```

Sino

```
// Disminuir el contador de intentos
```

```
intentos_restantes = intentos_restantes - 1
```

```
// Dar una pista al usuario
```

Si adivinanza > numero\_secreto Entonces

Escribir "El número es menor. Intentos restantes: ", intentos\_restantes

Sino

Escribir "El número es mayor. Intentos restantes: ", intentos\_restantes

FinSi

```
// Verificar si se han agotado los intentos
```

Si intentos\_restantes = 0 Entonces



Escribir "Lo siento, has agotado todos los intentos. El número era: ",

numero\_secreto

FinSi

FinSi

FinMientras

FinProceso

```

1  Proceso AdivinaElNumero
2  //40. Adivina el número: Genera un número aleatorio y pide al usuario que
3  //adivine el número. Utiliza un bucle while para repetir la solicitud hasta
4  //que adivine correctamente.
5  Definir numero_secreto, intento, intentos_restantes Como Entero
6  // Definir número aleatorio entre 1 y 100
7  numero_secreto = Azar(100)
8  // Visualizar intentos restantes
9  intentos_restantes = 3
10 // Bucle para los intentos
11 Mientras intentos_restantes > 0 Hacer
12     // Pide al usuario que adivine
13     Escribir "Adivina el número entre 1 y 100: "
14     Leer adivinanza
15     // Verificar si la adivinanza es correcta
16     Si adivinanza = numero_secreto Entonces
17         Escribir "¡Correcto! Has adivinado el número."
18         // Salir del bucle si la adivinanza es correcta
19         FinSi
20     Sino
21         // Disminuir el contador de intentos
22         intentos_restantes = intentos_restantes - 1
23         // Dar una pista al usuario
24         Si adivinanza > numero_secreto Entonces
25             Escribir "El número es menor". Intentos restantes: ", intentos_restantes
26         Sino
27             Escribir "El número es mayor". Intentos restantes: ", intentos_restantes
28         FinSi
29     // Verificar si se han agotado los intentos
30     Si intentos_restantes = 0 Entonces
31         Escribir "Lo siento, has agotado todos los intentos. El número era: ", numero_secreto
32         FinSi
33     FinMientras
34 FinProceso
    
```

\*\*\* Ejecución Iniciada. \*\*\*  
 Adivina el número entre 1 y 100:  
 > 1  
 El número es mayor. Intentos restantes: 2  
 Adivina el número entre 1 y 100:  
 > 3  
 El número es mayor. Intentos restantes: 1  
 Adivina el número entre 1 y 100:  
 > 4  
 El número es mayor. Intentos restantes: 0  
 Lo siento, has agotado todos los intentos. El número era: 46  
 \*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//42. Suma de números impares: Utiliza un bucle while para calcular la

//suma de los números impares del 1 al 100.

suma\_impar<-0

n<- 1

Mientras n <=100 Hacer

si n mod 2 <> 0 Entonces

suma\_impar<- suma\_impar + n

FinSi

n <- n+1

Fin Mientras

Escribir "La suma de los números impares del 1 al 100 es:", suma\_impar

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //42. Suma de números impares: Utiliza un bucle while para calcular la
3  //suma de los números impares del 1 al 100.
4  suma_impar<-0
5  n<- 1
6  Mientras n <=100 Hacer
7      si n mod 2 <> 0 Entonces
8          suma_impar<- suma_impar + n
9      FinSi
10     n <- n+1
11 Fin Mientras
12 + Escribir "La suma de los números impares del 1 al 100 es:", suma_impar
13 FinAlgoritmo
    
```

\*\*\* Ejecución Iniciada. \*\*\*  
 La suma de los números impares del 1 al 100 es:2500  
 \*\*\* Ejecución Finalizada. \*\*\*

Algoritmo sin\_titulo

//43. Contador de caracteres: Escribir un programa que lea una palabra y

//presenta cuantos caracteres hay en dicha palabra.

definir palabra Como Caracter

definir log\_palabra Como Entero

Escribir "Ingrese la palabra"

leer palabra

log\_palabra <- Longitud(palabra)

Escribir "La longitud de la palabra es " ,log\_palabra

FinAlgoritmo

```

1 Algoritmo sin_titulo
2 //43. Contador de caracteres: Escribir un programa que lea una palabra y
3 //presenta cuantos caracteres hay en dicha palabra.
4 definir palabra Como Caracter
5 definir log_palabra Como Entero
6 Escribir "Ingrese la palabra"
7 leer palabra
8 log_palabra <- Longitud(palabra)
9 Escribir "La longitud de la palabra es " ,log_palabra
10 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese la palabra
> hola
La longitud de la palabra es 4
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//44. Suma de números: Pide al usuario que ingrese números enteros

//positivos uno por uno y utiliza un bucle while para calcular la suma de

//estos números. El ciclo debe terminar cuando el usuario ingrese un número negativo.

definir numeros\_enteros , suma Como Entero

suma <-0

Mientras numeros\_enteros >= 0 Hacer

suma <- suma + numeros\_enteros

Escribir "Ingrese otro número entero positivo (ingrese un número negativo para

terminar):"

Leer numeros\_enteros

Fin Mientras

Escribir " la suma de los numeros es " , suma

FinAlgoritmo

```

1 Algoritmo sin_titulo
2 //44. Suma de números: Pide al usuario que ingrese números enteros
3 //positivos uno por uno y utiliza un bucle while para calcular la suma de
4 //estos números. El ciclo debe terminar cuando el usuario ingrese un número negativo.
5 definir numeros_enteros , suma Como Entero
6 suma <-0
7 Mientras numeros_enteros >= 0 Hacer
8 suma <- suma + numeros_enteros
9 Escribir "Ingrese otro número entero positivo (ingrese un número negativo para terminar):"
10 Leer numeros_enteros
11 Fin Mientras
12 Escribir " la suma de los numeros es " , suma
13 FinAlgoritmo

```

PSeInt - Ejecutando proceso SIN\_TITULO

```

*** Ejecución Iniciada. ***
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> 2
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> 3
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> 1
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> 4
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> 12
Ingrese otro número entero positivo (ingrese un número negativo para term
inar):
> -1
la suma de los numeros es 22
*** Ejecución Finalizada. ***

```

Algoritmo sin\_titulo

//45. Cuenta regresiva: Pide al usuario que ingrese un número entero

//positivo y utiliza un bucle while para mostrar una cuenta regresiva desde

//ese número hasta 1.

Definir n Como Entero

Escribir "Ingrese un número entero positivo:"

Leer n

Si n <= 0 Entonces

Escribir "El número ingresado no es válido. Debe ser un número entero positivo."

Sino

Escribir "Cuenta regresiva desde", n, "hasta 1:"

Mientras n >= 1 Hacer

Escribir n

n <- n - 1

Fin Mientras

Fin Si

FinAlgoritmo

```

1  Algoritmo sin_titulo
2  //45. Cuenta regresiva: Pide al usuario que ingrese un número entero
3  //positivo y utiliza un bucle while para mostrar una cuenta regresiva desde
4  //ese número hasta 1.
5
6  Definir n Como Entero
7  Escribir "Ingrese un número entero positivo:"
8  Leer n
9  Si n ≤ 0 Entonces
10     Escribir "El número ingresado no es válido. Debe ser un número entero positivo."
11     Sino
12         Escribir "Cuenta regresiva desde", n, "hasta 1:"
13         Mientras n ≥ 1 Hacer
14             Escribir n
15             n ← n - 1
16         Fin Mientras
17     Fin Si
18 FinAlgoritmo
  
```

PSeInt - Ejecutando proceso SIN\_TITULO  
 \*\*\* Ejecución Iniciada. \*\*\*  
 Ingrese un número entero positivo:  
 > 15  
 Cuenta regresiva desde15hasta 1:  
 15  
 14  
 13  
 12  
 11  
 10  
 9  
 8  
 7  
 6  
 5  
 4  
 3  
 2  
 1  
 \*\*\* Ejecución Finalizada. \*\*\*

Algoritmo ARREGLOS

Proceso SumaDeElementos

//46 Suma de elementos: Crea un arreglo de números enteros y

//calcula la suma de todos sus elementos.

// Definir variables

Definir n, i, suma Como Entero

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Inicializar la suma a 0

suma = 0

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

Escribir "Ingresa el elemento ", i, ": "

Leer numeros[i]

FinPara

// Calcular la suma de todos los elementos del arreglo

Para i = 1 Hasta n Con Paso 1 Hacer

suma = suma + numeros[i]

FinPara

// Mostrar la suma de los elementos

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

FinProceso

```

1  Proceso SumaDeElementos
2  //46 Suma de elementos: Crea un arreglo de números enteros y
3  //calcula la suma de todos sus elementos.
4  // Definir variables
5  Definir n, i, suma Como Entero
6  Definir numeros Como Entero
7  // Pedir al usuario que ingrese el tamaño del arreglo
8  Escribir "Ingresa el tamaño del arreglo: "
9  Leer n
10 // Declarar el arreglo con el tamaño ingresado por el usuario
11 Dimension numeros[n]
12 // Inicializar la suma a 0
13 suma = 0
14 // Llenar el arreglo con números enteros ingresados por el usuario
15 Para i = 1 Hasta n Con Paso 1 Hacer
16     Escribir "Ingresa el elemento ", i, ": "
17     Leer numeros[i]
18 FinPara
19 // Calcular la suma de todos los elementos del arreglo
20 Para i = 1 Hasta n Con Paso 1 Hacer
21     suma = suma + numeros[i]
22 FinPara
23 // Mostrar la suma de los elementos
24 Escribir "La suma de todos los elementos es: ", suma
25 FinProceso
    
```

Console Output:

```

PSeint - Ejecutando proceso SUMADEELEMENTOS
*** Ejecución Iniciada. ***
Ingresa el tamaño del arreglo:
> 2
Ingresa el elemento 1:
> 1
Ingresa el elemento 2:
> 1
La suma de todos los elementos es: 2
*** Ejecución Finalizada. ***
    
```

Proceso PromedioDeCalificaciones

//47. Promedio de calificaciones: Crea un arreglo de calificaciones

//(números decimales) y calcula el promedio de las calificaciones.

// Definir variables

Definir n, i Como Entero

Definir suma, promedio Como Real

Definir calificaciones Como Real

// Pedir al usuario que ingrese el número de calificaciones

Escribir "Ingresa el número de calificaciones: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension calificaciones[n]

// Inicializar la suma a 0

suma = 0

// Llenar el arreglo con calificaciones ingresadas por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

Escribir "Ingresa la calificación ", i, ": "

Leer calificaciones[i]

FinPara

// Calcular la suma de todas las calificaciones

Para i = 1 Hasta n Con Paso 1 Hacer

suma = suma + calificaciones[i]

FinPara

// Calcular el promedio de las calificaciones

promedio = suma / n

// Mostrar el promedio de las calificaciones

Escribir "El promedio de las calificaciones es: ", promedio

FinProceso

```

1  Proceso PromedioDeCalificaciones
2  //47. Promedio de calificaciones: Crea un arreglo de calificaciones
3  // (números decimales) y calcula el promedio de las calificaciones.
4
5  Definir n, i Como Entero
6  Definir suma, promedio Como Real
7  Definir calificaciones Como Real
8  // Pedir al usuario que ingrese el número de calificaciones
9  Escribir "Ingresa el número de calificaciones: "
10 Leer n
11 // Declarar el arreglo con el tamaño ingresado por el usuario
12 Dimension calificaciones[n]
13 // Inicializar la suma a 0
14 suma = 0
15 // Llenar el arreglo con calificaciones ingresadas por el usuario
16 Para i = 1 Hasta n Con Paso 1 Hacer
17     Escribir "Ingresa la calificación ", i, ": "
18     Leer calificaciones[i]
19 FinPara
20 // Calcular la suma de todas las calificaciones
21 Para i = 1 Hasta n Con Paso 1 Hacer
22     suma = suma + calificaciones[i]
23 FinPara
24 // Calcular el promedio de las calificaciones
25 promedio = suma / n
26 // Mostrar el promedio de las calificaciones
27 Escribir "El promedio de las calificaciones es: ", promedio
28 FinProceso
    
```

PSeInt - Ejecutando proceso PROMEDIODECALIFICACIONES

```

*** Ejecución Iniciada. ***
Ingresa el número de calificaciones:
> 2
Ingresa la calificación 1:
> 20
Ingresa la calificación 2:
> 20
El promedio de las calificaciones es: 20
*** Ejecución Finalizada. ***
    
```

☐ No cerrar esta ventana ☐ Siempre visible

Proceso MayorYMenorValor

//48. Buscar un elemento: Pide al usuario que ingrese un número y verifica si ese número está

// presente en un arreglo dado.

Definir n, i, maximo, minimo Como Entero

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

Escribir "Ingresa el elemento ", i, ": "

Leer numeros[i]

FinPara

// Inicializar maximo y minimo con el primer elemento del arreglo

maximo = numeros[1]

minimo = numeros[1]

// Encontrar el valor máximo y mínimo

Para i = 2 Hasta n Con Paso 1 Hacer

Si numeros[i] > maximo Entonces

maximo = numeros[i]

FinSi

Si numeros[i] < minimo Entonces

    minimo = numeros[i]

FinSi

FinPara

// Mostrar el valor máximo y mínimo

Escribir "El valor máximo es: ", maximo

Escribir "El valor mínimo es: ", minimo

FinProceso

```

1  Proceso MayorMenorValor
2  //49. Buscar un elemento: Pide al usuario que ingrese un número y verifica si ese número está
3  //presente en un arreglo dado.
4  Definir n, i, maximo, minimo Como Entero
5  Definir numeros Como Entero
6  // Pedir al usuario que ingrese el tamaño del arreglo
7  Escribir "Ingresa el tamaño del arreglo: "
8  Leer n
9  // Declarar el arreglo con el tamaño ingresado por el usuario
10 Dimension numeros[n]
11 // Llenar el arreglo con números enteros ingresados por el usuario
12 Para i = 1 Hasta n Con Paso 1 Hacer
13     Escribir "Ingresa el elemento ", i, ": "
14     Leer numeros[i]
15 FinPara
16 // Inicializar maximo y minimo con el primer elemento del arreglo
17 maximo = numeros[1]
18 minimo = numeros[1]
19 // Encontrar el valor máximo y mínimo
20 Para i = 2 Hasta n Con Paso 1 Hacer
21     Si numeros[i] > maximo Entonces
22         maximo = numeros[i]
23     FinSi
24     Si numeros[i] < minimo Entonces
25         minimo = numeros[i]
26     FinSi
27 FinPara
28 // Mostrar el valor máximo y mínimo
29 Escribir "El valor máximo es: ", maximo
30 Escribir "El valor mínimo es: ", minimo
31 FinProceso
  
```

\*\*\* Ejecución Iniciada. \*\*\*  
 Ingresa el tamaño del arreglo:  
 > 14  
 Ingresa el elemento 1:  
 > 2  
 Ingresa el elemento 2:  
 > 4  
 Ingresa el elemento 3:  
 > 3  
 Ingresa el elemento 4:  
 > 2  
 Ingresa el elemento 5:  
 > 5  
 Ingresa el elemento 6:  
 > 6  
 Ingresa el elemento 7:  
 > 7  
 Ingresa el elemento 8:  
 > 12  
 Ingresa el elemento 9:  
 > 100  
 Ingresa el elemento 10:  
 > 23  
 Ingresa el elemento 11:  
 > 12  
 Ingresa el elemento 12:  
 > 15  
 Ingresa el elemento 13:  
 > 11  
 Ingresa el elemento 14:  
 > 1000  
 El valor máximo es: 1000  
 El valor mínimo es: 2  
 \*\*\* Ejecución Finalizada. \*\*\*

Proceso BuscarElementoEnArreglo

//49. Buscar un elemento: Pide al usuario que ingrese un número y

//verifica si ese número está presente en un arreglo dado.

Definir n, i, numero\_a\_buscar Como Entero

Definir encontrado Como Logico

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

    Escribir "Ingresa el elemento ", i, ": "

    Leer numeros[i]

FinPara

// Pedir al usuario que ingrese el número a buscar

Escribir "Ingresa el número a buscar: "

Leer numero\_a\_buscar

// Inicializar la variable encontrado a Falso

```

encontrado = Falso

// Buscar el número en el arreglo

Para i = 1 Hasta n Con Paso 1 Hacer

    Si numeros[i] = numero_a_buscar Entonces

        encontrado = Verdadero

    FinSi

FinPara

// Mostrar el resultado de la búsqueda

Si encontrado Entonces

    Escribir "El número ", numero_a_buscar, " está presente en el arreglo."

Sino

    Escribir "El número ", numero_a_buscar, " no se encuentra en el arreglo."

FinSi

FinProceso
    
```

## Proceso ContarElementosPares

```

//50. Contar elementos pares: Cuenta cuántos números pares hay en un
//arreglo de números enteros.
    
```

```

Definir n, i, contador_pares Como Entero

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

    Escribir "Ingresa el elemento ", i, ": "

    Leer numeros[i]

FinPara
    
```



```
// Inicializar el contador de números pares a 0

contador_pares = 0

// Contar los números pares en el arreglo

Para i = 1 Hasta n Con Paso 1 Hacer

    Si numeros[i] % 2 = 0 Entonces

        contador_pares = contador_pares + 1

    FinSi

FinPara

// Mostrar el número de elementos pares

Escribir "El número de elementos pares en el arreglo es: ", contador_pares

FinProceso
```

```
1 Proceso ContarElementosPares
2 //50. Contar elementos pares: Cuenta cuántos números pares hay en un
3 //arreglo de números enteros.
4 Definir n, i, contador_pares Como Entero
5 Definir numeros Como Entero
6 // Pedir al usuario que ingrese el tamaño del arreglo
7 Escribir "Ingresa el tamaño del arreglo: "
8 Leer n
9 // Declarar el arreglo con el tamaño ingresado por el usuario
10 Dimension numeros[n]
11 // Llenar el arreglo con números enteros ingresados por el usuario
12 Para i = 1 Hasta n Con Paso 1 Hacer
13     Escribir "Ingresa el elemento ", i, ": "
14     Leer numeros[i]
15 FinPara
16 // Inicializar el contador de números pares a 0
17 contador_pares = 0
18 // Contar los números pares en el arreglo
19 Para i = 1 Hasta n Con Paso 1 Hacer
20     Si numeros[i] % 2 = 0 Entonces
21         contador_pares = contador_pares + 1
22     FinSi
23 FinPara
24 // Mostrar el número de elementos pares
25 Escribir "El número de elementos pares en el arreglo es: ", contador_pares
26 FinProceso
```

PSeInt - Ejecutando proceso CONTARELEMENTOSPARES

```
*** Ejecución Iniciada. ***
Ingresa el tamaño del arreglo:
> 4
Ingresa el elemento 1:
> 12
Ingresa el elemento 2:
> 13
Ingresa el elemento 3:
> 15
Ingresa el elemento 4:
> 17
El número de elementos pares en el arreglo es: 1
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

## Proceso InvertirArreglo

```
//51. Inversión de un arreglo: Invierte el orden de los elementos

//en un arreglo. Por ejemplo, [1, 2, 3] se convierte en [3, 2, 1].

// Definir variables

Definir n, i, temp Como Entero

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

    Escribir "Ingresa el elemento ", i, ": "

    Leer numeros[i]

FinPara

// Invertir el orden de los elementos del arreglo

Para i = 1 Hasta n / 2 Con Paso 1 Hacer
```

```
temp = numeros[i]

numeros[i] = numeros[n - i + 1]

numeros[n - i + 1] = temp
```

FinPara

// Mostrar el arreglo invertido

Escribir "El arreglo invertido es: "

Para i = 1 Hasta n Con Paso 1 Hacer

Escribir numeros[i]

FinPara

FinProceso

```
1 Proceso InvertirArreglo
2 //51. Inversión de un arreglo: Invierte el orden de los elementos
3 //en un arreglo. Por ejemplo, [1, 2, 3] se convierte en [3, 2, 1].
4 //Definir variables
5 Definir n, i, temp Como Entero
6 Definir numeros Como Entero
7 // Pedir al usuario que ingrese el tamaño del arreglo
8 Escribir "Ingresa el tamaño del arreglo: "
9 Leer n
10 // Declarar el arreglo con el tamaño ingresado por el usuario
11 Dimension numeros[n]
12 // Llenar el arreglo con números enteros ingresados por el usuario
13 Para i = 1 Hasta n Con Paso 1 Hacer
14   Escribir "Ingresa el elemento ", i, ": "
15   Leer numeros[i]
16 FinPara
17 // Invertir el orden de los elementos del arreglo
18 Para i = 1 Hasta n / 2 Con Paso 1 Hacer
19   temp = numeros[i]
20   numeros[i] = numeros[n - i + 1]
21   numeros[n - i + 1] = temp
22 FinPara
23 // Mostrar el arreglo invertido
24 Escribir "El arreglo invertido es: "
25 Para i = 1 Hasta n Con Paso 1 Hacer
26   Escribir numeros[i]
27 FinPara
28 FinProceso
```

\*\*\* Ejecución Iniciada. \*\*\*  
 Ingresa el tamaño del arreglo:  
 > 3  
 Ingresa el elemento 1:  
 > 2  
 Ingresa el elemento 2:  
 > 12  
 Ingresa el elemento 3:  
 > 3  
 El arreglo invertido es:  
 3  
 12  
 2  
 \*\*\* Ejecución Finalizada. \*\*\*

Proceso BuscarIndiceEnArreglo

```
//52. Buscar el índice: Pide al usuario que ingrese un valor y
//encuentra el índice de ese valor en un arreglo. Si el valor aparece más
//de una vez, muestra todos los índices.
```

Definir n, i, valor\_a\_buscar Como Entero

Definir encontrado Como Logico

Definir numeros Como Entero

// Pedir al usuario que ingrese el tamaño del arreglo

Escribir "Ingresa el tamaño del arreglo: "

Leer n

// Declarar el arreglo con el tamaño ingresado por el usuario

Dimension numeros[n]

// Llenar el arreglo con números enteros ingresados por el usuario

Para i = 1 Hasta n Con Paso 1 Hacer

Escribir "Ingresa el elemento ", i, ": "

Leer numeros[i]

FinPara

// Pedir al usuario que ingrese el valor a buscar

Escribir "Ingresa el valor a buscar: "

Leer valor\_a\_buscar

// Inicializar la variable encontrado a Falso

encontrado = Falso

// Buscar el valor en el arreglo y mostrar los índices donde se encuentra

Para i = 1 Hasta n Con Paso 1 Hacer

Si numeros[i] = valor\_a\_buscar Entonces

encontrado = Verdadero

Escribir "El valor ", valor\_a\_buscar, " se encuentra en el índice ", i

FinSi

FinPara

// Si no se encuentra el valor, mostrar un mensaje

Si No encontrado Entonces

Escribir "El valor ", valor\_a\_buscar, " no se encuentra en el arreglo."

FinSi

FinProceso

```
1  Proceso BuscarIndiceEnArreglo
2  //Si: Buscar el índice. Pide al usuario que ingrese un valor y
3  //encuentra el índice de ese valor en un arreglo. Si el valor aparece más
4  //de una vez, muestra todos los índices.
5  Definir n, i, valor_a_buscar Como Entero
6  Definir encontrado Como Logico
7  Definir numeros Como Entero
8  // Pide al usuario que ingrese el tamaño del arreglo
9  Escribir "Ingresa el tamaño del arreglo: "
10 Leer n
11 // Declarar el arreglo con el tamaño ingresado por el usuario
12 Dimensionar numeros[n]
13 // Llenar el arreglo con números enteros ingresados por el usuario
14 Para i = 1 Hasta n Con Paso 1 Hacer
15     Escribir "Ingresa el elemento ", i, ": "
16     Leer numeros[i]
17 FinPara
18 // Pide al usuario que ingrese el valor a buscar
19 Escribir "Ingresa el valor a buscar: "
20 Leer valor_a_buscar
21 // Inicializar la variable encontrado a Falso
22 encontrado = Falso
23 // Buscar el valor en el arreglo y mostrar los índices donde se encuentra
24 Para i = 1 Hasta n Con Paso 1 Hacer
25     Si numeros[i] = valor_a_buscar Entonces
26         encontrado = Verdadero
27         Escribir "El valor ", valor_a_buscar, " se encuentra en el índice ", i
28     FinSi
29 FinPara
30 // Si no se encuentra el valor, mostrar un mensaje
31 Si No encontrado Entonces
32     Escribir "El valor ", valor_a_buscar, " no se encuentra en el arreglo."
33 FinSi
34 FinProceso
```

\*\*\* Ejecución Iniciada. \*\*\*  
Ingresa el tamaño del arreglo:  
> 3  
Ingresa el elemento 1:  
> 12  
Ingresa el elemento 2:  
> 13  
Ingresa el elemento 3:  
> 1  
Ingresa el valor a buscar:  
> 1  
El valor 1 se encuentra en el índice 3  
\*\*\* Ejecución Finalizada. \*\*\*

☐ No cerrar esta ventana ☐ Siempre visible

// Función saludo con parámetro nombre

Funcion saludo(nombre)

Escribir "Hola que tal ", nombre

FinFuncion

// Función suma con parámetros a y b

Funcion suma(a, b)

Definir resultado Como Entero

resultado <- a + b

Escribir "La suma de los números es ", resultado

FinFuncion

// Función multiplicar con parámetros n1 y n2

Funcion multiplicar(n1, n2)

Definir resultado Como Entero

resultado <- n1 \* n2

Escribir "La multiplicación de los números es ", resultado

FinFuncion

// Función numerospares con parámetro numero

Funcion numerospares(numero)

Si numero MOD 2 = 0 Entonces

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

Sino

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

Fin Si

FinFuncion

// Función area\_rectangulo con parámetros base y altura

Funcion area\_rectangulo(base, altura)

Definir area Como Entero

area <- base \* altura

Escribir "El área del rectángulo es ", area

FinFuncion

// Función imprimir\_nombre con parámetro nombre

Funcion imprimir\_nombre(nombre)

Escribir "Hola, mi nombre es ", nombre

FinFuncion

// Función contar\_caracter con parámetros frase y caracter\_a\_contar

Funcion contar\_caracter(frase, caracter\_a\_contar)

Definir l, pos, contador Como Entero

l <- Longitud(frase)

contador <- 0

Para pos <- 0 Hasta l - 1 Con Paso 1 Hacer

Si Subcadena(frase, pos, pos) = caracter\_a\_contar Entonces

contador <- contador + 1

FinSi

FinPara

Escribir "El carácter ", caracter\_a\_contar, " aparece ", contador, " veces en la frase."

FinFuncion

```
// Función imprimir_numeros sin parámetros
```

```
Funcion imprimir_numeros(rango)
```

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

```
        Escribir i
```

```
    Fin Para
```

```
FinFuncion
```

```
//convertir de celcius
```

```
Funcion celsius_a_fahrenheit(celsius)
```

```
    Definir fahrenheit como real
```

```
    fahrenheit <- (celsius * 9/5) + 32
```

```
    Escribir "La temperatura" ,celsius,"en grados Fahrenheit es: ", fahrenheit FinFuncion
```

```
//Contar caracteres de una frase
```

```
Funcion contar_caracteres(frase)
```

```
    Definir long Como Entero
```

```
    long <- Longitud(frase)
```

```
    Escribir "La cantidad de caracteres en la frase es: ", long FinFuncion
```

```
Algoritmo funciones
```

```
    // Instanciación de las funciones con parámetros saludo("Juan")
```

```
    suma(5, 7)
```

```
    multiplicar(3, 4)
```

```
    numerospares(6)
```

```
    area_rectangulo(4, 5)
```

```
    imprimir_nombre("María")
```

```
    contar_caracter("Hola mundo", "o")
```

```
    imprimir_numeros(10)
```

```
        celsius_a_fahrenheit(100)
```

```
        contar_caracteres('hola mundo')
```

```
FinAlgoritmo
```