

SOFTWARE REVIEW

PSEUDOCÓDIGO –

ESTRUCTURAS REPETITIVAS

Resuelva los siguientes desafíos creando el pseudocódigo con la herramienta PSeInt.

1. ESTRUCTURA REPETITIVA: PARA (FOR)

Ejercicio 1: Imprimir números pares

Escribe un programa que imprima todos los **números pares** del 2 al 20 en la consola.

• Ejemplo 1:

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 2, 4, 6, 8, 10, 12, 14, 16, 18, 20.
- **Explicación:** El programa recorre los números del 2 al 20 y muestra solo los que son pares.

• Ejemplo 2:

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 2, 4, 6, 8, 10, 12, 14, 16, 18, 20.
- **Explicación:** Los números pares se imprimen en orden ascendente.

```
Proceso PARA_serie_pares
    Para i<-1 Hasta 10 Con Paso 1 Hacer
        pares=i*2
        Escribir pares
    Fin Para
FinProceso
```

Ejercicio 2: Suma de múltiplos de 3

Escribe un programa que calcule la **suma** de todos los múltiplos de 3 entre 1 y 30.

• **Ejemplo 1:**

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 165.
- **Explicación:** Se suman los números múltiplos de 3 (3, 6, 9, ..., 30).

Ejemplo 2:

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 165.
- **Explicación:** La suma acumulativa de los múltiplos de 3 entre 1 y 30 da como resultado 165.

```
Proceso PARA_suma_multiplos_3
    suma=0 //variable acumuladora
    Para i<-1 Hasta 30 Con Paso 1 Hacer
        Si i%3==0 Entonces
            suma=suma+i
        Fin Si
    Fin Para
    Escribir suma
FinProceso
```

Ejercicio 3: Factorial de un número

Escribe un programa que pida al usuario un número entero positivo y calcule su factorial.

• **Ejemplo 1:**

- **Entrada:** 5.
- **Salida:** 120.
- **Explicación:** $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.

• **Ejemplo 2:**

- **Entrada:** 3.
- **Salida:** 6.
- **Explicación:** $3! = 3 \times 2 \times 1 = 6$.

```
Proceso PARA_suma_multiplos_5
```

```

    Escribir 'Ingrese el número del que desea hallar su
factorial'
    Leer num
    multi=1 //variable acumuladora
    Para i<-1 Hasta num Con Paso 1 Hacer
        multi=multi*i
    Fin Para
    Escribir multi
FinProceso

```

Ejercicio 4: Generar serie Fibonacci

Escribe un programa que genere los primeros 10 números de la **serie de Fibonacci**.

• Ejemplo 1:

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.
- **Explicación:** La serie de Fibonacci comienza con 0 y 1, y cada número subsiguiente es la suma de los dos anteriores.

• Ejemplo 2:

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.
- **Explicación:** La salida es la secuencia de Fibonacci hasta el décimo número.

```

Proceso PARA_serie_Fibonacci
    num1=0
    num2=1
    Escribir num1
    Escribir num2
    Para i<-1 Hasta 8 Con Paso 1 Hacer
        suma=num1+num2
        Escribir suma
        num1=num2
        num2=suma
    Fin Para
FinProceso

```

Ejercicio 5: Generar triángulo de Pascal

Escribe un programa que genere el **Triángulo de Pascal** hasta la fila n , donde n es un número ingresado por el usuario. El Triángulo de Pascal es un arreglo triangular donde el número en cada posición es la suma de los dos números directamente encima de él en la fila anterior. **Nota:** No usar vectores

• **Ejemplo 1:**

- **Entrada:** $n = 5$.
- **Salida:**

Copiar código

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

- **Explicación:** El programa genera el Triángulo de Pascal hasta la quinta fila.

• **Ejemplo 2:**

- **Entrada:** $n = 7$.
- **Salida:**

Copiar código

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

- **Explicación:** El programa construye el Triángulo de Pascal hasta la séptima fila.

2. ESTRUCTURA REPETITIVA: MIENTRAS (WHILE)

Ejercicio 1: Contar hasta 10

Escribe un programa que cuente desde 1 hasta 10, mostrando cada número en la consola.

• **Ejemplo 1:**

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
- **Explicación:** El programa utiliza un bucle que se repite mientras el contador sea menor o igual a 10.

• **Ejemplo 2:**

- **Entrada:** N/A (No se requiere entrada).
- **Salida:** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
- **Explicación:** El contador comienza en 1 y se incrementa hasta llegar a 10.

```
Proceso MIENTRAS_cuenta_10
    num=1
    suma=0
    Mientras suma<10 Hacer
        suma=num+1
        Escribir suma
        num=suma
    Fin Mientras
FinProceso
```

Ejercicio 2: Verificación de contraseña

Crea un programa que solicite al usuario ingresar una contraseña y continúe solicitándola hasta que el usuario la ingrese correctamente.

• **Ejemplo 1:**

- **Entrada:** Contraseña correcta: "abc123".
- **Salida:** "Contraseña correcta".
- **Explicación:** El usuario ingresa la contraseña "abc123", y el programa termina.

• **Ejemplo 2:**

- **Entrada:** Contraseña incorrecta: "abc", "123", correcta: "abc123".
- **Salida:** "Contraseña incorrecta", "Contraseña incorrecta", "Contraseña correcta".
- **Explicación:** El usuario necesita intentar varias veces antes de ingresar la contraseña correcta.

```
Proceso MIENTRAS_password
    correct='abc123'
    switch=Falso
```

```

Mientras switch==Falso Hacer
    Escribir 'Ingrese la contraseña'
    Leer password
    Si password==correct Entonces
        Escribir 'Contraseña correcta'
        switch=Verdadero
    SiNo
        Escribir 'Contraseña incorrecta'
    Fin Si
Fin Mientras
FinProceso

```

Ejercicio 3: Sumar dígitos de un número

Escribe un programa que sume los dígitos de un número entero positivo ingresado por el usuario.

• Ejemplo 1:

- **Entrada:** 1234.
- **Salida:** 10.
- **Explicación:** La suma de los dígitos $1+2+3+4$ es 10.

• Ejemplo 2:

- **Entrada:** 567.
- **Salida:** 18.
- **Explicación:** La suma de los dígitos $5+6+7$ es 18.

```

Proceso MIENTRAS_suma_digitos
    Escribir 'Ingrese un número'
    Leer num
    sum=0
    digit=0
    Mientras num>0 Hacer
        digit=num mod 10
        num=trunc(num/10)
        sum=sum+digit
    Fin Mientras
    Escribir sum
FinProceso

```

Ejercicio 4: Invertir un número

Crea un programa que pida al usuario un número entero positivo y lo imprima en orden inverso.

- **Ejemplo 1:**

- **Entrada:** 1234.
- **Salida:** 4321.
- **Explicación:** El número ingresado se invierte completamente.

- **Ejemplo 2:**

- **Entrada:** 8901.
- **Salida:** 1098.
- **Explicación:** El número se muestra en orden inverso al original.

```
Proceso MIENTRAS_digitos_invertidos
    Escribir 'Ingrese un número'
    Leer num
    Mientras num>0 Hacer
        Escribir num mod 10
        num=trunc(num/10)
    Fin Mientras
FinProceso
```

Ejercicio 5: Calcular raíz cuadrada con método de aproximación (Dificultad Alta)

Escribe un programa que calcule la **raíz cuadrada** de un número utilizando el método de aproximaciones sucesivas.

- **Ejemplo 1:**

- **Entrada:** 16.
- **Salida:** Aproximadamente 4.0.
- **Explicación:** El programa encuentra la raíz cuadrada de 16 por aproximación.

- **Ejemplo 2:**

- **Entrada:** 25.

- **Salida:** Aproximadamente 5.0.
- **Explicación:** El programa encuentra la raíz cuadrada de 25 utilizando el mismo método.

3. ESTRUCTURA REPETITIVA: REPITA HASTA (DO-WHILE)

Ejercicio 1: Leer número positivo

Escribe un programa que pida al usuario ingresar un número positivo y siga solicitándolo hasta que el número ingresado sea mayor que 0.

• Ejemplo 1:

- **Entrada:** -5, -2, 3.
- **Salida:** "Número válido".
- **Explicación:** El programa sigue solicitando un número hasta que el usuario ingrese un valor positivo.

• Ejemplo 2:

- **Entrada:** -1, 7.
- **Salida:** "Número válido".
- **Explicación:** El primer número es inválido, pero el segundo número es aceptado.

Ejercicio 2: Adivinar un número

Escribe un programa que solicite al usuario ingresar números hasta que la suma acumulada de los números pares ingresados sea mayor o igual a 50. El programa debe ignorar los números impares.

• Ejemplo 1:

- **Entrada:** 8, 15, 12, 5, 30.
- **Salida:** "Suma de números pares: 50".
- **Explicación:** El programa suma los números 8, 12 y 30 (totalizando 50) y se detiene.

• Ejemplo 2:

- **Entrada:** 4, 10, 3, 7, 20, 8.

- **Salida:** "Suma de números pares: 42", "Suma de números pares: 60".
- **Explicación:** El programa primero suma 4, 10 y 20 (42) y luego, al añadir 8, la suma llega a 60, superando los 50.

Ejercicio 3: Solicitar contraseña

Escribe un programa que solicite al usuario una contraseña y continúe solicitándola hasta que se ingrese una contraseña que tenga al menos 8 caracteres.

• Ejemplo 1:

- **Entrada:** "abc123", "abcdefg8".
- **Salida:** "Contraseña muy corta", "Contraseña válida".
- **Explicación:** La primera contraseña tiene menos de 8 caracteres, la segunda es aceptada.

• Ejemplo 2:

- **Entrada:** "password1".
- **Salida:** "Contraseña válida".
- **Explicación:** La contraseña ingresada cumple con el requisito de longitud.

Ejercicio 4: Suma de números hasta 100

Crea un programa que permita al usuario ingresar números y siga sumándolos hasta que la suma acumulada sea mayor que 100.

• Ejemplo 1:

- **Entrada:** 25, 30, 50.
- **Salida:** "Suma total: 105".
- **Explicación:** La suma de los números ingresados excede 100, por lo que el programa termina.

• Ejemplo 2:

- **Entrada:** 40, 40, 30.
- **Salida:** "Suma total: 110".
- **Explicación:** El programa se detiene cuando la suma acumulada supera

los 100.

Ejercicio 5: Adivinar la contraseña con intentos limitados (Dificultad Alta)

Escribe un programa que solicite al usuario una contraseña y le permita 3 intentos para adivinarla. Si falla los 3 intentos, el programa debe terminar.

• Ejemplo 1:

- **Entrada:** Contraseña correcta: "java123", intentos: "python321", "java123".
- **Salida:** "Contraseña incorrecta", "Contraseña correcta".
- **Explicación:** El usuario acierta en el segundo intento.

• Ejemplo 2:

- **Entrada:** Contraseña correcta: "pass123", intentos: "12345", "admin", "pass123".
- **Salida:** "Contraseña incorrecta", "Contraseña incorrecta", "Contraseña correcta".
- **Explicación:** El usuario acierta en el tercer intento.

```
Proceso HASTA_contraseña
    correct='campus2024'
    switch=Falso
    attempts=0
    Repetir
        Escribir 'Ingresa la contraseña'
        Leer password
        attempts=attempts+1
        Si password=correct Entonces
            Escribir 'Contraseña correcta'
            switch=Verdadero
        SiNo
            Escribir 'Contraseña incorrecta'
        FinSi
    Hasta Que switch=Verdadero o attempts=3
FinProceso
```