

**“Año de la Unidad, la Paz y el Desarrollo”**

**CURSO: PRINCIPIOS DE ALGORITMOS (44557)**

**DOCENTE: FERNANDO TOMÁS VILLANUEVA**

**CICLO II**

**TAREA:**

**S12 - Resolver ejercicios**

**INTEGRANTES:**

<b>JOSE LUIS OLGUIN BARRUETO</b>	<b>U23229256</b>
<b>ROBERTO AGUSTÍN MEJÍA COLLAZOS</b>	<b>U23254461</b>
<b>MANUEL ÁNGEL PECHO SANTOS</b>	<b>U23201694</b>
<b>MIGUEL ANGEL VELASQUEZ YSUIZA</b>	<b>U23231519</b>
<b>RAFAEL FERNANDO QUISPE MONTALVAN</b>	<b>U23209464</b>
<b>MARIELENA CARDENAS FLORES</b>	<b>U23330326</b>

## Separata de trabajo 09 – Estructuras repetitivas PARA

Semana 12

### Ejercicios propuestos

1. Realizar un pseudocódigo que imprima los números impares entre 0 y 50

The image displays three windows related to a programming exercise:

- Pseudocode Editor:** Contains the following code:

```
1 Proceso Impares
2   Para i<=1 Hasta 50 Con Paso 2 Hacer
3     Escribir i
4   FinPara
5 FinProceso
6
```
- Console Window (PSeInt):** Shows the execution output:

```
*** Ejecución Iniciada. ***
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
*** Ejecución Finalizada. ***
```
- Flowchart (PSDraw):** A diagram titled "Proceso Impares" showing a loop structure. It starts with an oval labeled "i" containing "1", "2", and "50". An arrow points to a decision diamond labeled "i". From the diamond, an arrow points down to an oval labeled "FinProceso". Another arrow points from the diamond back to the start of the loop.

2. Construir un programa que pida ingresar un número, luego muestre la tabla de multiplicar de dicho número del 1 al 12.

```

1 Proceso Tabla_de_Multiplicar
2 Definir num, cont, res Como Entero
3 cont ← 1
4 Escribir 'Escribir numero'
5 Leer num
6 Mientras cont ≤ 12 Hacer
7     res ← num * cont
8     Escribir num, ' * ', cont, ' = ', res
9     cont ← cont + 1
10 FinMientras
11 FinProceso
12

```

PSDraw - Tabla\_de\_Multiplicar

```

graph TD
    Start([Proceso Tabla_de_Multiplicar]) --> Define[Definir num, cont, res...]
    Define --> Init[cont ← 1]
    Init --> Write[Escribir numero]
    Write --> Read[/num/]
    Read --> Decision{cont ≤ 12}
    Decision -- V --> Calc[res ← num * cont]
    Calc --> Write2[/num, ' * ', cont, ' = .../]
    Write2 --> Inc[cont ← cont + 1]
    Inc --> Decision
    Decision -- F --> End([FinProceso])

```

PSInt - Ejecutando proceso TAB...

```

*** Ejecución Iniciada. ***
Escribir numero
> 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
5 * 11 = 55
5 * 12 = 60
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible

3. Desarrollar un pseudocódigo que luego de ingresar 2 números naturales imprima los números naturales que hay entre ambos empezando por el menor. Asumimos que el usuario ingresa dos números diferentes y el segundo es mayor que el primero.

```

1  Proceso Numeros_Naturales
2  Definir n1, n2, i, j Como Entero
3  Escribir 'Ingrese primer numero'
4  Leer n1
5  Escribir 'Ingrese segundo numero'
6  Leer n2
7  i ← n1+1
8  j ← n2
9  Mientras i<j Hacer
10     Escribir i
11     i ← i+1
12  FinMientras
13 FinProceso
14

```

```

*** Ejecución Iniciada. ***
Ingrese primer numero
> 5
Ingrese segundo numero
> 10
6
7
8
9
*** Ejecución Finalizada. ***

```

PSDraw - Numeros\_Naturales

```

graph TD
    Start([Inicio]) --> Def[Definir n1, n2, i, j C...]
    Def --> In1[/Ingrese primer numero/]
    In1 --> n1[n1]
    n1 --> In2[/Ingrese segundo numero/]
    In2 --> n2[n2]
    n2 --> i1[i ← n1+1]
    i1 --> j1[j ← n2]
    j1 --> Cond{i < j}
    Cond -- V --> Out[i]
    Out --> Inc[i ← i+1]
    Inc --> Cond
    Cond -- F --> End([FinProceso])

```

4. Calcular el precio total de una compra de “n” productos (el valor “n” se ingresa por teclado) si se ingresa el precio de cada producto y la cantidad comprada de este.

```

1 Proceso Precio_Total
2   Definir n, cant Como Entero
3   Definir pre, pretot Como Real
4   pretot ← 0
5   Escribir 'Ingresar cantidad de productos'
6   Leer n
7   Para i←1 Hasta n Con Paso 1 Hacer
8     Escribir 'Ingresar la cantidad del producto', i
9     Leer cant
10    Escribir 'Ingresar el precio del producto', i
11    Leer pre
12    pretot ← pretot+(cant*pre)
13  FinPara
14  Escribir 'El precio total de la compra es de : ', pretot
15 FinProceso
16

```

```

*** Ejecución Iniciada. ***
Ingresar cantidad de productos
> 2
Ingresar la cantidad del producto1
> 5
Ingresar el precio del producto1
> 0.70
Ingresar la cantidad del producto2
> 3
Ingresar el precio del producto2
> 3.50
El precio total de la compra es de : 14
*** Ejecución Finalizada. ***

```

```

graph TD
    Start([Proceso Precio_Total]) --> Def1[Definir n, cant Como E...]
    Def1 --> Def2[Definir pre, pretot Co...]
    Def2 --> Init[pretot ← 0]
    Init --> Input1[/Ingresar cantidad de .../]
    Input1 --> Dec1{n}
    Dec1 --> Input2[/Ingresar la cantidad .../]
    Input2 --> Dec2{cant}
    Dec2 --> Input3[/Ingresar el precio de.../]
    Input3 --> Dec3{pre}
    Dec3 --> Calc[pretot ← pretot+(cant*pre)]
    Calc --> Print[/El precio total de la.../]
    Print --> Dec1
    Dec1 --> End([FinProceso])

```