

LABORATORIO N°4

PROGRAMACIÓN DE COMPUTADORES

Objetivos del Laboratorio:

Al término de la clase los alumnos serán capaces de:

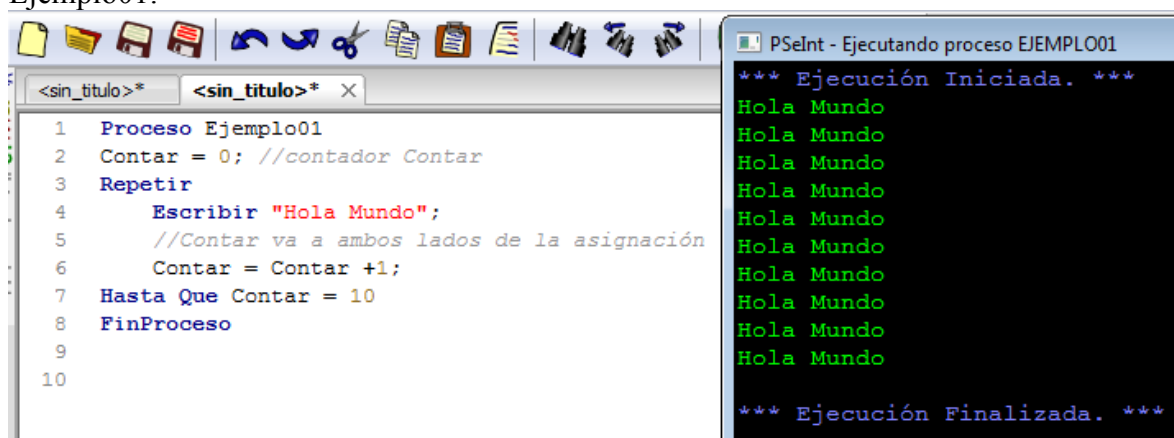
- Diseñar algoritmos no secuenciales definiendo las entradas, el proceso y la salida que dará solución a un determinado problema.
- Comprender las estructuras de control iterativas o cíclicas
Repetir-Hasta Que,
Mientras-Hacer-FinMientras
- Entender la función de un **contador** y un **sumador**
- Probar algoritmos utilizando diversas estructuras de control con la ayuda de **PSeInt**

Conceptos:

¿Qué es un contador?

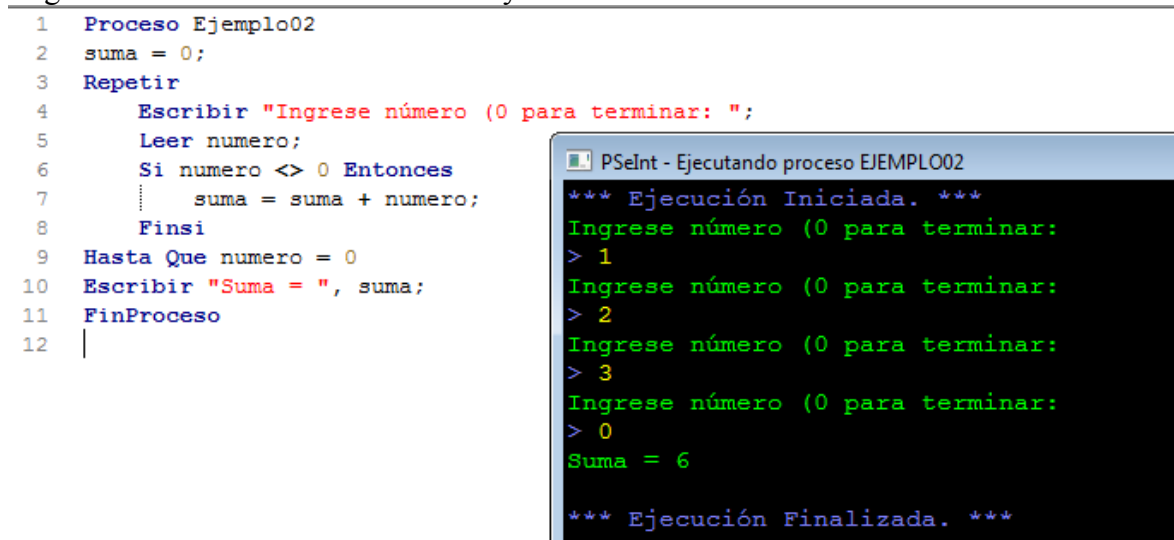
Un **contador** es una variable cuyo valor se incrementa o decrementa en una cantidad constante, cada vez que se produce un determinado suceso o iteración. Los contadores se usan para contar sucesos o iteraciones de un ciclo. Se debe asignar un valor inicial antes del ciclo y se incrementen o decrementan dentro del ciclo. El valor inicial es aquel desde el cual se quiere comenzar a contar.

Ejemplo01:



¿Qué es un sumador?

Un **sumador** es una variable que suma sobre sí misma un conjunto de valores. La variable está en ambos lados de la asignación. En cada iteración se le suma un valor a la variable. Se debe asignar un valor inicial antes del ciclo y se incrementen del ciclo.



Important Hint:

La diferencia fundamental entre un **contador** y un **sumador** (acumulador) es la siguiente: el contador aumenta (o disminuye) una cantidad fija en cada ciclo. En cambio, el sumador aumenta o disminuye una cantidad variable en cada ciclo.

Actividades:

1.- Ejercicio: Diseñe un algoritmo que permita a una persona ingresar tantos números como desee y al final obtenga como resultados la suma de los números, la cantidad de números y el promedio de ellos.

Para el ejercicio se pide:

- Analizar el problema
- Describir las Entradas, Proceso y Salida
- Diseñar el algoritmo a través de pseudocódigo o diagrama de flujo
- Ejecutar y probar el algoritmo

ENTRADAS: El usuario ingresa tantos números como desee. Por lo tanto, no se sabe cuántos números ingresará. Entonces, la entrada son los números que ingresa el usuario. Se necesita una variable para almacenar el o los números que ingresa el usuario.

SALIDAS: La suma y el promedio de los números.

PROCESO: Se requiere para el diseño:

- un sumador para sumar (acumular) los números ingresados
- un contador para contar la cantidad de números ingresados
- un ciclo Repetir-Hasta Que se ingrese el número 0, indicando que no se quiere seguir ingresando números.

Solución en pseudocódigo PSeInt.

Proceso Lab0401

```

suma ← 0;
contador ← 0;
Repetir
    Leer numero;
    Si numero <> 0
        suma ← suma + numero;
        contador ← contador + 1;
    FinSi
Hasta Que n = 0
Si contador <> 0
    promedio ← suma/contador;
FinSi
Escribir suma, contador, promedio;

```

FinProceso

2.- Ejercicio: Modifique el algoritmo anterior para obtener:

- La suma, cantidad y promedio de números pares
- La suma, cantidad y promedio de números impares

3.- Ejercicio: Diseñe un algoritmo que sume todos los números enteros positivos anteriores a un número n dado por el usuario. Por ejemplo, si el usuario ingresa 5 el programa debería entregar 10, ($1+2+3+4 = 10$).

Para el ejercicio se pide:

- Analizar el problema
- Describir las Entradas, Proceso y Salida
- Diseñar el algoritmo a través de pseudocódigo o diagrama de flujo
- Ejecutar y probar el algoritmo

ENTRADAS: El usuario ingresa un único número entero mayor o igual a cero, que llamaremos numero.

SALIDAS: La suma de los números enteros positivo menor que numero

PROCESO: Se requiere para el diseño:

- un sumador para sumar los números desde el 1 al numero - 1 que llamaremos suma.
- una variable, que puede partir en 1, para controlar que el proceso no supere el número ingresado. Lo llamaremos numeroDeVeces.
- un ciclo **Repetir-Hasta** que se ejecuta 1 o más veces, dependiendo del número ingresado.
- analizar el caso cuando se ingresa 1. ¿qué debería entregar como resultado?

Solución en pseudocódigo PSeInt.

Proceso Lab0403

```

suma = 0;
veces = 0;
Leer numero;
Repetir
    suma= suma + veces;
    veces = veces + 1;
Hasta Que numero = veces
Escribir suma;

```

FinProceso

4.- Ejercicio: Modifique el algoritmo anterior para obtener el promedio de los números sumados.

5.- Ejercicio: Resuelva el Ejercicio 3 usando la instrucción *Mientras-Que*

6.- Ejercicio: Se requiere calcular el valor de la siguiente serie:

$$y = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

n es un número entero mayor que cero ingresado por el usuario.

Para el ejercicio se pide:

- Analizar el problema
- Describir las Entradas, Proceso y Salida
- Diseñar el algoritmo a través de pseudocódigo o diagrama de flujo utilizando el ciclo *Repetir-Hasta Que*. para validar el n. Es decir, repetir hasta que se ingrese un valor correcto.
- Diseñe su algoritmo, en diagrama de flujo y pseudocódigo, utilizando el ciclo *Mientras-Hacer* para obtener la suma.
- Ejecutar y probar el algoritmo

7.- Ejercicio: Se necesita diseñar un simulador del llenado de un estanque que tiene una capacidad máxima definida por el usuario. Cada vez es posible agregar 0.5 litro al estanque. El usuario controla la cantidad de litros que desea echar en el estanque. Diseñe un algoritmo que permita echar una cantidad de litros en el estanque y que muestre permanentemente el nivel que lleva (la cantidad de litros).

Para el ejercicio se pide:

- Analizar el problema
- Describir las Entradas, Proceso y Salida
- Diseñar el algoritmo a través de pseudocódigo o diagrama de flujo utilizando el ciclo *Repetir-Hasta Que*.
- Diseñe su algoritmo, en diagrama de flujo y pseudocódigo, utilizando el ciclo *Mientras-Hacer*
- Ejecutar y probar el algoritmo

8.- Ejercicio: Diseñar un algoritmo que genere una sucesión de números de rango x y final n partiendo desde cero. Por ejemplo, si el rango es 3 y el final es 13, la sucesión que genera es: **0,3,6,9,12**

Para el ejercicio se pide:

- Analizar el problema
- Describir las Entradas, Proceso y Salida
- Diseñar el algoritmo a través de pseudocódigo o diagrama de flujo utilizando el ciclo *Repetir-Hasta Que*.
- Diseñe su algoritmo, en diagrama de flujo y pseudocódigo, utilizando el ciclo *Mientras-Hacer*
- Ejecutar y probar el algoritmo