



## PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL FORMATO GUÍA DE APRENDIZAJE

- DENOMINACIÓN DEL PROGRAMA DE FORMACIÓN:** Programación de Software
- CÓDIGO DEL PROGRAMA DE FORMACIÓN:** 2712557
- NOMBRE DEL PROYECTO:**
- FASE DEL PROYECTO:**
- ACTIVIDAD DE PROYECTO:** Análisis y comprensión de herramientas tecnológicas necesarias para el manejo de la información en la empresa
- COMPETENCIA DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO**
- RESULTADOS DE APRENDIZAJE ALCANZAR:** RAP 1. RESOLVER PROCESOS LÓGICOS A TRAVÉS DE LA IMPLEMENTACIÓN DE ALGORITMOS Y EL LENGUAJE DE PROGRAMACIÓN SELECCIONADO.
- DURACIÓN DE LA GUÍA:** 18 horas

### 2. Presentación

Las computadoras se usan a menudo para repetir miles de cálculos sin cometer errores, por esta y muchas otras razones, es de esperar que existan dentro de los lenguajes de programación estructuras que permitan realizar tareas repetidamente.

La ejecución repetida de un conjunto de sentencias (instrucciones) se llama iteración.

Los lenguajes de programación proporcionan varias características para realizarla, en pseudocódigo vamos a tratar tres de estas estructuras:

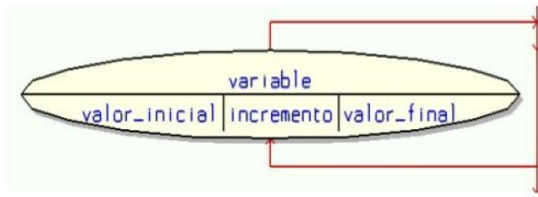
#### Para

La estructura para permite realizar una tarea cíclica un número fijo de veces. En pseudocódigo tiene la siguiente forma:

```
Para variable_numerica <- valor_inicial Hasta valor_final Con Paso paso Hacer  
| secuencia_de_acciones  
Fin Para
```

- La **asignación\_de\_inicio** es una asignación a una variable que controlará la ejecución del ciclo por ejemplo  $i=0$
- El **valor\_final** es una expresión aritmética que indica el número máximo de iteraciones El incremento es el valor en el cual se incrementará la variable en cada ejecución

El siguiente diagrama de flujo muestra gráficamente un ciclo para:



El siguiente ejemplo define una variable de nombre *i* de tipo entero que inicia en 1 y que repite la escritura de un número en la pantalla 5 veces

```
Algoritmo escribiri
  Definir i Como Entero

  Para i=1 Hasta 5 Con Paso 1 Hacer
    Escribir i
  Fin Para
FinAlgoritmo
```

```
PSeInt - Ejecutando proceso ESCRIBIRI

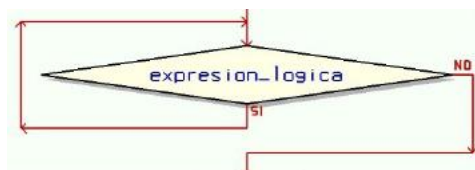
1
2
3
4
5
*** Ejecución Finalizada. ***
```

## Mientras

La estructura mientras permite realizar una tarea cíclica, mientras que una condición lógica se cumpla, esto permite una mayor flexibilidad. En pseudocódigo tiene la siguiente estructura:

```
Mientras expresion_logica Hacer
  secuencia_de_acciones
Fin Mientras
```

El siguiente diagrama de flujo muestra de manera gráfica el flujo de las instrucciones dentro del ciclo mientras





El siguiente ejemplo define una variable de nombre *i* de tipo entero que inicia en 1 y que repite la escritura de un número en la pantalla 5 veces:

```
Algoritmo escribiri
  Definir i Como Entero
  i=1
  Mientras i≤5 Hacer
    escribir i
    i=i+1
  Fin Mientras
FinAlgoritmo
```

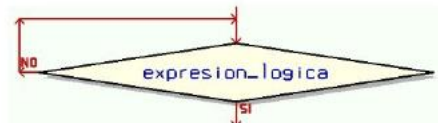
```
PSeInt - Ejecutando proceso ESCRIBIRI
*** Ejecución Iniciada. ***
1
2
3
4
5
*** Ejecución Finalizada. ***
```

## Repetir Hasta que

La estructura repetir hasta que permite realizar una tarea cíclica, similar a la que se realiza en el ciclo mientras, pero con la peculiaridad de que se siempre se ejecuta por lo menos una vez la primera instrucción. En pseudocódigo tiene la siguiente estructura:

```
Repetir
  secuencia_de_acciones
Hasta Que expresion_logica
```

En un diagrama de flujo tiene la siguiente representación:



El siguiente ejemplo define una variable de nombre *i* de tipo entero que inicia en 1 y que repite la escritura de un número en la pantalla 5 veces, note que a diferencia del ciclo mientras en esta estructura se ejecuta la instrucción Escribir *i* por lo menos una vez, sin hacer la evaluación de la condición lógica.



Algoritmo sin\_titulo

Definir i Como Entero

i=1

Repetir

Escribir i

i ← i + 1

Hasta Que i>5

FinAlgoritmo

PSeInt - Ejecutando proceso SIN\_...

\*\*\* Ejecución Iniciada. \*\*\*

1

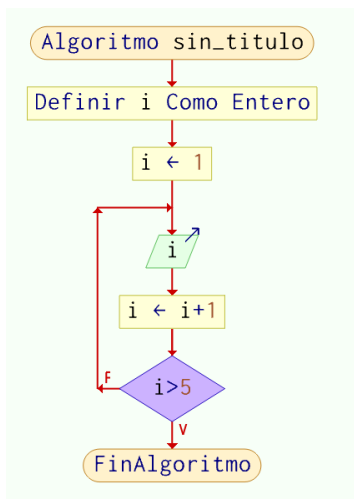
2

3

4

5

\*\*\* Ejecución Finalizada. \*\*\*



### Recomendaciones para elegir un CICLO:

Al elegir el ciclo adecuado en un programa, es importante comprender la lógica y los requisitos del problema que estás resolviendo. Aquí hay algunas pautas generales para ayudarte a decidir qué ciclo utilizar:

1. Si sabes de antemano cuántas veces se repetirá un bloque de código, puedes usar el ciclo "para". Este ciclo es útil cuando conoces el número exacto de iteraciones.
2. Si no sabes de antemano cuántas veces se repetirá un bloque de código, pero tienes una condición que determina cuándo se debe detener el ciclo, puedes usar el ciclo "mientras". Este ciclo se ejecutará siempre que la condición sea verdadera.
3. Si necesitas ejecutar al menos una vez un bloque de código y luego evaluar una condición para determinar si se repite nuevamente, puedes usar el ciclo "repetir-hasta que". Este ciclo se ejecutará al menos una vez y continuará repitiéndose hasta que la condición sea verdadera.



4. Si necesitas repetir un bloque de código y quieres tener la opción de interrumpir la repetición en cualquier momento, puedes usar el ciclo "hacer-mientras". Este ciclo se ejecutará al menos una vez y luego verificará una condición para decidir si se repite nuevamente.

En última instancia, la elección del ciclo dependerá de la lógica y los requisitos específicos del problema que estás abordando. Es importante entender la estructura y el comportamiento de cada ciclo para seleccionar el más adecuado para tu situación. Además, ten en cuenta que en algunos casos puedes utilizar diferentes ciclos para lograr el mismo resultado, por lo que también es posible adaptar y ajustar los ciclos según tus necesidades.

### **ACTIVIDADES PROPUESTAS**

Estimado Aprendiz, realice el pseudocódigo en Pseint guarde en carpeta:

 Semana 3: Ciclos

1. Realizar un programa que muestre los números del 1 al 10 en pantalla.
2. Solicitar al usuario un número y mostrar en pantalla todos los números pares desde 1 hasta ese número.
3. Solicitar al usuario un número y mostrar en pantalla la suma de todos los números impares desde 1 hasta ese número.
4. Solicitar al usuario un número y mostrar en pantalla la tabla de multiplicar de ese número del 1 al 10.
5. Solicitar al usuario un número y mostrar en pantalla todos los números primos desde 1 hasta ese número.
6. Solicitar al usuario un número y mostrar en pantalla la suma de todos los números desde 1 hasta ese número.
7. Solicitar al usuario un número y mostrar en pantalla la suma de los dígitos de ese número.
8. Realizar un programa que muestre los números impares del 1 al 20 en pantalla.
9. Solicitar al usuario un número y mostrar en pantalla si es un número primo.
10. Realizar un programa que muestre la siguiente serie: 1, 4, 9, 16, 25, ..., 100.
11. Diseñe un algoritmo representándolo en DF y Pseudocódigo tal que ingresados 100 números naturales, permita determinar y mostrar:
  - a. ¿Cuántos son menores que 15?
  - b. ¿Cuántos son mayores que 50?



c. ¿Cuántos están comprendidos entre 25 y 45?

d. ¿Cuál es el promedio de los 100 números?

El algoritmo debe considerar las validaciones de datos que sean necesarias.

1. Un número abundante es todo número entero positivo que cumple que la suma de sus divisores propios es mayor que el propio número. Por ejemplo, 12 es abundante ya que sus divisores propios son 1, 2, 3, 4 y 6 y se cumple que  $1+2+3+4+6=16$ , que es mayor que 12. Diseñe un algoritmo representándolo en DF y Pseudocódigo que verifique si un cierto número entero positivo es abundante. El algoritmo debe considerar las validaciones de datos que sean necesarias.

## EJERCICIOS CICLOS CON PARA (FOR)

Estimado Aprendiz, realice el pseudocódigo en Pseint guarde en carpeta:



Ciclos



Mientras



Repetir



Para

1. **Calcular la suma de los primeros N números enteros:** Realiza un programa que pida al usuario un número entero N y calcule la suma de los primeros N números enteros ( $1 + 2 + 3 + \dots + N$ ).
2. **Mostrar los números pares del 1 al 100:** Crea un programa que imprima en pantalla todos los números pares del 1 al 100.
3. **Tabla de multiplicar:** Diseña un programa que solicite un número entero y luego muestre su tabla de multiplicar del 1 al 10.
4. **Factorial de un número:** Escribe un programa que pida al usuario un número entero y calcule su factorial.
5. **Suma de números pares e impares:** Realiza un programa que lea 10 números (uno por uno) y calcule la suma de los números pares y la suma de los números impares.
6. **Promedio de calificaciones:** Crea un programa que lea las calificaciones de 5 estudiantes (una por una) y calcule el promedio de estas calificaciones.
7. **Secuencia Fibonacci:** Escribe un programa que pida un número entero N y genere los primeros N términos de la secuencia de Fibonacci.
8. **Número de divisores:** Realiza un programa que pida un número entero y determine cuántos divisores tiene.
9. **Contar letras en una cadena:** Crea un programa que solicite al usuario una cadena de texto y cuente cuántas veces aparece una letra específica en dicha cadena.
10. **Potencias de 2:** Diseña un programa que imprima las primeras N potencias de 2 ( $2^1, 2^2, 2^3, \dots, 2^N$ ).
11. **Promedio de N números:** Escribe un programa que pida al usuario un número N y luego solicite N números (uno por uno), calculando y mostrando el promedio de esos números.
12. **Suma de dígitos:** Crea un programa que pida un número entero y calcule la suma de sus dígitos.
13. **Mostrar números del 100 al 1:** Diseña un programa que imprima en pantalla los números del 100 al 1 en orden descendente.
14. **Contar números positivos, negativos y ceros:** Escribe un programa que lea 10 números (uno por uno) y cuente cuántos son positivos, cuántos son negativos y cuántos son ceros.
15. **Sumar múltiplos de 3 y 5:** Realiza un programa que calcule la suma de todos los números múltiplos de 3 y 5 que hay entre 1 y 100.

## EJERCICIOS EXTRAS UTILIZANDO LA ESTRUCTURA DE CICLOS EN PSEINT

Estimado aprendiz, estos ejercicios simulan situaciones cotidianas en las que se requiere utilizar ciclos para procesar datos o realizar cálculos.

1. Leer y sumar los gastos diarios de una persona hasta que se ingrese un valor negativo como indicador de fin de ingreso de gastos. Al finalizar, mostrar el total de gastos.
2. Solicitar repetidamente el ingreso de números positivos y calcular su promedio. El proceso debe detenerse cuando se ingrese un número negativo.
3. Leer una serie de números que representan las edades de los miembros de una familia. Calcular la edad promedio y determinar cuántos miembros tienen una edad superior al promedio.
4. Leer una serie de calificaciones de un examen hasta que se ingrese un valor negativo como indicador de fin de ingreso de calificaciones. Calcular el promedio de las calificaciones y mostrar si el estudiante aprobó o reprobó (asumiendo que la nota de aprobación es 60).
5. Solicitar repetidamente el ingreso de palabras hasta que se ingrese la palabra "fin". Contar la cantidad total de palabras ingresadas.
6. Leer una serie de números que representan las temperaturas registradas durante una semana. Calcular la temperatura máxima y mínima de la semana.
7. Solicitar repetidamente el ingreso de nombres de empleados y sus respectivos salarios hasta que se ingrese el nombre "fin" como indicador de fin de ingreso. Calcular el salario promedio de los empleados ingresados y mostrar los nombres de aquellos cuyo salario sea superior al promedio.
8. Leer una serie de números que representan las ventas diarias de una tienda. Calcular el total de ventas y mostrar el día en que se obtuvo la mayor venta.
9. Solicitar repetidamente el ingreso de números positivos y calcular su suma. Detener el proceso cuando la suma supere un valor predefinido.
10. Leer una serie de números y contar cuántos de ellos son múltiplos de 5.



## EJERCICIOS EXTRAS UTILIZANDO LA ESTRUCTURA DE CICLOS EN PSEINT

Estimado aprendiz, estos ejercicios simulan situaciones cotidianas en las que se requiere utilizar ciclos para procesar datos o realizar cálculos.

1. Leer y sumar los gastos diarios de una persona hasta que se ingrese un valor negativo como indicador de fin de ingreso de gastos. Al finalizar, mostrar el total de gastos.
2. Solicitar repetidamente el ingreso de números positivos y calcular su promedio. El proceso debe detenerse cuando se ingrese un número negativo.
3. Leer una serie de números que representan las edades de los miembros de una familia. Calcular la edad promedio y determinar cuántos miembros tienen una edad superior al promedio.
4. Leer una serie de calificaciones de un examen hasta que se ingrese un valor negativo como indicador de fin de ingreso de calificaciones. Calcular el promedio de las calificaciones y mostrar si el estudiante aprobó o reprobó (asumiendo que la nota de aprobación es 60).
5. Solicitar repetidamente el ingreso de palabras hasta que se ingrese la palabra "fin". Contar la cantidad total de palabras ingresadas.
6. Leer una serie de números que representan las temperaturas registradas durante una semana. Calcular la temperatura máxima y mínima de la semana.
7. Solicitar repetidamente el ingreso de nombres de empleados y sus respectivos salarios hasta que se ingrese el nombre "fin" como indicador de fin de ingreso. Calcular el salario promedio de los empleados ingresados y mostrar los nombres de aquellos cuyo salario sea superior al promedio.
8. Leer una serie de números que representan las ventas diarias de una tienda. Calcular el total de ventas y mostrar el día en que se obtuvo la mayor venta.
9. Solicitar repetidamente el ingreso de números positivos y calcular su suma. Detener el proceso cuando la suma supere un valor predefinido.
10. Leer una serie de números y contar cuántos de ellos son múltiplos de 5.