



Carrera: Licenciatura y Profesorado en Matemática
Guía de Trabajos Prácticos N° 3 – CICLOS: Estructuras Repetitivas

Objetivos:

- ✓ Que el alumno incorpore los conceptos de Ciclos.
- ✓ Identifique y aplique las distintas estructuras propuestas.
- ✓ Entienda y utilice la terminología técnica apropiada.
- ✓ Pueda avanzar en forma independiente según sus tiempos de aprendizajes con respecto a los nuevos conceptos.
- ✓ Desarrolle las actividades propuestas en tiempo y forma.

Hasta aquí hemos hecho programas que solo se repiten una vez, pero en la programación necesitamos que los programas corran varias veces y que nos presenten información al final de correr varias veces, en estos casos usaremos **ciclos**, que son estructuras de repetición, que *iteran* hasta cumplir con una condición o simplemente indicamos cuantas veces se van a repetir.

Nota: Para evitar ambigüedades, todos los ciclos deben cerrarse siempre, no es posible que hayan "Ciclos abiertos"



Ciclo Repetir Mientras:

El ciclo mientras se utiliza cuando se quiere ejecutar repetidamente un bloque de instrucciones basado en una condición, el ciclo se repite mientras la condición se cumple.

Mientras <condición> **Hacer**
 <Accion1>
 <Accion2>
 <.....>
FinMientras



**Ejemplo:**

Se le da al usuario de nuestro algoritmo 10 intentos para adivinar un número que es generado aleatoriamente, utilizar la estructura mientras para verificar si el usuario acierta el número o se le agotaron los intentos.

// Juego simple que pide al usuario que adivine un numero en 10 intentos

Algoritmo Adivina_Numero

Definir intentos,num_secreto,num_ingresado **Como Entero**

intentos<-10

num_secreto <- azar(100)+1

Escribir "Adivine el numero (de 1 a 100):"

Leer num_ingresado

Mientras num_secreto<>num_ingresado Y intentos>1 **Hacer**

Si num_secreto>num_ingresado **Entonces**

Escribir "Muy bajo"

SiNo

Escribir "Muy alto"

FinSi

 intentos <- intentos-1

Escribir "Le quedan ",intentos," intentos:"

Leer num_ingresado

FinMientras

Si num_secreto=num_ingresado **Entonces**

Escribir "Exacto! Usted adivino en ",11-intentos," intentos."

SiNo

Escribir "El numero era: ",num_secreto

FinSi

FinAlgoritmo**Ejercicio 3.1**

Realizar un algoritmo que efectúe la suma de los números positivos pares siempre y cuando estos números sean menores a un número ingresado previamente por teclado.

Ejercicio 3.2

Desarrollar un algoritmo que sea capaz de calcular y mostrar las longitudes de n circunferencias, mientras el radio ingresado de la circunferencia sea mayor a cero.

Ejercicio 3.3

Se desea diseñar un algoritmo que, de acuerdo a la altura de la persona, le permita entrar a un juego en un parque de diversiones. En este caso, para poder subirse a la montaña rusa, si la persona mide 1.30 mts. o más puede ingresar.

El algoritmo debe finalizar cuando se ingresa un valor 0 (cero) para la altura de la persona.

Ejercicio 3.4

Diseñe un algoritmo que solicite a un usuario, que desea iniciar sesión en un sitio web, nombre y contraseña mientras ambos no sean correctos.



Ejercicio 3.5

Escribir un algoritmo que permita ingresar 10 números y al final muestre la suma de los números ingresados.

Ejercicio 3.6

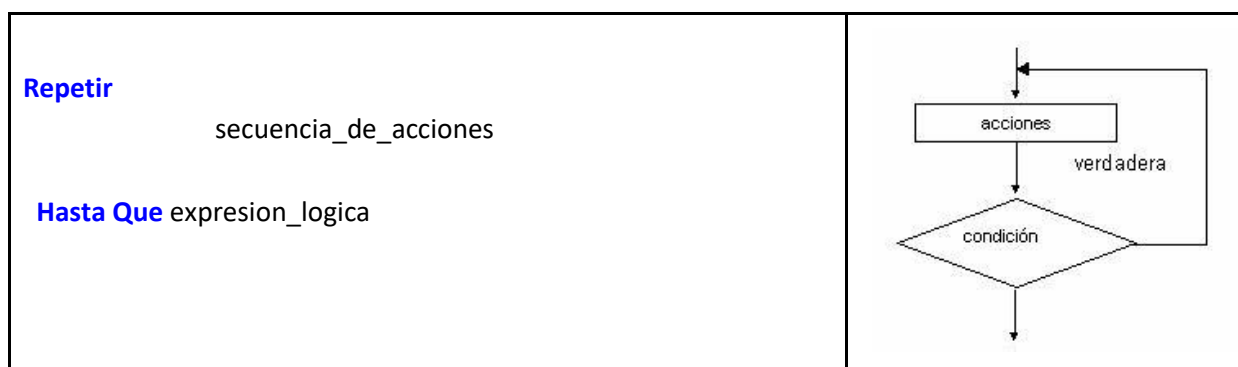
Escribir un algoritmo que permita ingresar las notas de N alumnos para calcular e informar al final el promedio de estas notas.

Ejercicio 3.7

Se pretende simular el funcionamiento de una caja de supermercado, por lo tanto usando una respuesta ("S" o "N") para controlar la salida del programa, escribir un algoritmo que permita ingresar, por cada producto; el código, el precio, cantidad y luego calcular el subtotal, iva y total a pagar por cada producto. Al final mostrar el monto total del ticket.

Estructuras Repetitivas Repetir hasta que Ciclo Repetir hasta que

Descripción: Este ciclo hace que se repita ese mismo ciclo las veces que sea necesario hasta que la condición sea verdadera, en otros lenguajes se llamaría **Do While**.



Ejemplo :

Un algoritmo que se finalice cuando un numero sea menor o igual a 5. Tenemos que utilizar operadores lógicos en este caso <=. Declarar una variable para que ingresemos un número por teclado. Inicializamos un ciclo repetir hasta que. Mostramos el resultado por pantalla.

Proceso Repetir_hasta_que

x<-0

Repetir

Escribir "Introduzca un numero del 1 al 5 : "

Leer x

Mostrar "X vale : ", x

Hasta Que x<=5

FinProceso



Podemos ver que es muy sencillo el programa entra al ciclo y te dirá que ingreses un número, pero ese número tiene que ser igual o menor que 5 por ejemplo 0,1,2,3,4 y 5, si no se repetirá hasta que ingresemos cualquiera de los números anteriores y se cerrara o finalizara el programa.

Ejercicios Ciclos Repetir hasta que:

Ejercicio 4.1.

Enunciado: Desarrollar un algoritmo que solicite que ingrese usuario y contraseña, Si el usuario ingresa bien sus credenciales le da un mensaje de bienvenida y si no que cuente la cantidad de veces que ingresa mal sus credenciales, cuando supere las tres entradas erróneas muestre un mensaje por pantalla “Usted ha superado los tres intentos válidos, su clave ha sido invalidada”, el algoritmo blanquea la clave del usuario.

Ejercicio 4.2.

Enunciado: desarrolle un algoritmo que calcula la superficie de un rectángulo hasta que la respuesta “¿desea continuar calculando la superficie de un rectángulo? (S/N)?” sea “N”

Ejercicio.4.3.

Enunciado: hacer un algoritmo que permita calcular la suma de pares de números. Pedirá dos números al usuario y mostrará su suma, volviendo a repetir hasta que ambos números introducidos sean 0.

Ejercicio.4.4

Enunciado: hacer un algoritmo que divida dos números que introduzca el usuario. Si el segundo número es cero, se le deberá avisar y volver a pedir tantas veces como sea necesario, hasta que introduzca un número distinto de cero, momento en que se calculará y mostrará el resultado de la división. Luego pregunta “¿desea dividiendo? (S/N)?” cuando la respuesta sea “N” sale.

Estructuras Repetitivas Para (for)

Ciclo Para

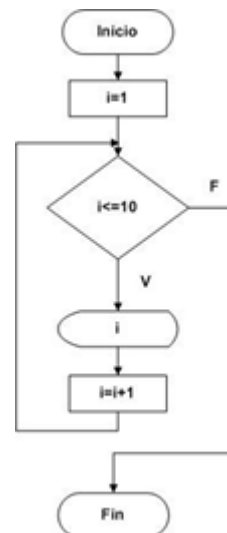
Sintaxis

Para variable <- valor_inicial **Hasta** valor_final **Con Paso** Paso **Hacer**
 <instrucciones>

FinPara



Para <variable> <- <valor_inicial> **Hasta** <valor_final> **Con Paso**
 <Paso> **Hacer**
 <Accion1>
 <Accion2>
 <.....>
FinPara



Descripción

El ciclo Para se utiliza generalmente para ejecutar un conjunto de instrucciones que se repiten un número de veces, establecido antes de ejecutar el ciclo. Variable: es de tipo entero
 Valor_inicial: este puede ser un número entero o una variable entera.

Valor_final: este puede ser un número entero o una variable entera.

Paso : este puede ser un número entero o una variable entera.

Ejercicios Ciclos Para:

Ejercicio 5.1.

Enunciado: Desarrollar un algoritmo que determine en un conjunto de diez números la cantidad de negativos, cuántos son mayores de 50 y cuántos están comprendidos entre 25 y 45.

Ejercicio 5.2.

Enunciado: Escribir un algoritmo que calcule la suma de los cuadrados de los 10 primeros números naturales.

Ejercicio 5.3.

Enunciado: Escribir un algoritmo que permita ingresar 10 números y calcule el promedio.

Ejercicio 5.4.

Enunciado: Escribir un algoritmo que calcule el factorial de un número.

El factorial de un número N se simboliza: N!, donde $N! = N * (N-1)!$.

Sólo existen factoriales de números naturales y cero. El factorial de $0! = 1$.

Ej: $5! = 5 * 4 * 3 * 2 * 1 * 1$.