



UNIVERSIDAD DE
MANIZALES



TÉCNICA PROFESIONAL EN CONFIGURACIÓN DE SERVICIOS PARA COMERCIO ELECTRÓNICO

MÓDULO: FUNDAMENTOS DE PROGRAMACIÓN

FACULTAD DE CIENCIAS E INGENIERÍA



UNIVERSIDAD DE
MANIZALES



FUNDAMENTOS DE PROGRAMACIÓN

Competencia académica:

- ✓ Crear estructuras de datos lógicas para la solución de problemas, teniendo en cuenta principios de comprensión y abstracción.

Manejo tecnológico:

- ✓ Utiliza en forma eficiente las herramientas necesarias para diagramar problemas a través de algoritmos.

Competencia Laboral General:

- ✓ Se comunicar efectivamente, utilizando todos los medios disponibles para ello, reconociendo y respetando las diferencias individuales.
- ✓ Demuestra integración y colaboración con los demás compañeros, para la consecución de objetivos comunes.



UNIVERSIDAD DE
MANIZALES



GUIA 2: ¿CÓMO DARLE INSTRUCCIONES AL COMPUTADOR?



INDICADORES DE DESEMPEÑO

CONCEPTUAL:

Reconoce el alcance de la programación para la solución de problemas en un mundo altamente digitalizado.

PROCEDIMENTAL:

- ✓ Aplica técnicas para la solución de problemas.
- ✓ Representa un problema a través de un algoritmo.



UNIVERSIDAD DE
MANIZALES



- ✓ Utiliza operadores aritméticos-lógicos y relacionales para la solución de problemas.
- ✓ Utiliza adecuadamente las expresiones computacionales en la solución de problemas.
- ✓ Elabora diagramas de datos Entrada –Proceso - Salida de Información.
- ✓ Ilustra la solución algorítmica mediante diagrama de flujo.
- ✓ Diagrama y testea las estructuras de decisión y de repetición en diagramas de flujo.
- ✓ Convierte diagramas de flujo a escritura de pseudocódigos.
- ✓ Convierte las estructuras de decisión y de repetición de diagramas de flujo a pseudocódigo.
- ✓ Realiza pruebas de Escritorio al Pseudocódigo.

ACTITUDINAL:

- ✓ Toma decisiones frente a la resolución de problemas y ejecución de actividades apoyado en sus conocimientos y habilidades.
- ✓ Demuestra integración y colaboración con los demás compañeros, para la consecución de objetivos comunes.

A. VIVENCIAS

TRABAJO INDIVIDUAL:

1. Con base en mis propios conocimientos, respondo las siguientes preguntas:
 - a. ¿Qué es un algoritmo?
 - b. ¿Qué es un diagrama de flujo?
 - c. ¿Cómo puedo representar un problema?
 - d. ¿Qué es un pseudocódigo?

TRABAJO GRUPAL:

2. Intercambio mis respuestas con un compañero y procedo a realizar la siguiente lectura, al terminar la lectura me podré dar cuenta si las respuestas dadas anteriormente fueron acertadas.

TRABAJO CON EL PROFESOR:

3. El profesor verifica el intercambio de respuestas.

B. FUNDAMENTACIÓN TEÓRICA

TRABAJO GRUPAL:

1. Procedo a realizar las siguientes lecturas, y tomo nota buscando las respuestas a las preguntas anteriores.

¿CÓMO DARLE INSTRUCCIONES A UN COMPUTADOR?

Los computadores son máquinas que realizan tareas a partir de las instrucciones que los humanos les damos y estas instrucciones deben ser claras, específicas, sin ambigüedades para que el computador las comprenda y las ejecute tal cual lo indicado.

¿QUÉ ES UN ALGORITMO?([link video](#))

Un algoritmo es un conjunto **finito** de **instrucciones**¹, **ordenadas** secuencialmente y **libres de ambigüedad**, que debe llevar a cabo un computador para lograr un resultado previsible ².

Las características fundamentales que debe cumplir todo algoritmo son:

finito: Tiene un inicio y un fin.

ordenadas Tiene un orden lógico secuencial para realizarse

¹ **Una instrucción** como una forma de enseñar impartiendo conocimientos o datos a una persona, animal o dispositivo tecnológico. En nuestro caso, como programadores, debemos darle instrucciones al computador o dispositivos tecnológico para que realicen las acciones que queremos.

² WordReference.com (2019). Definición de algoritmo. Disponible en: <http://www.wordreference.com/definicion/algoritmo>

libres de ambigüedad: instrucciones claras que no permitan confusión ni acciones alternas

¿CUÁLES SON LAS PARTES DE UN ALGORITMO?

Cualquier problema que involucre manejo de información requiere identificar las *entradas*, los *procesos* y las *salidas*.

Recordemos que en la sección anterior aprendimos las **partes de un problema** y los **pasos para solucionarlo**, ahora relacionemos estos conceptos con las partes de un algoritmo:

Entradas (analizar el problema, datos, recursos): Es la información que nos dan en el problema para iniciar su solución. Razón por la cual hay que leerlo para identificar los datos iniciales.

Procesos (planear, operaciones, ejecutar): Son las operaciones que se deben realizar para convertir los datos en salidas.

Estos procesos pueden ser:

- ✓ Operaciones matemáticas simples. [Ejemplo](#).
- ✓ [Operaciones que involucren condiciones](#). [Ejemplo](#).
- ✓ [Operaciones repetitivas](#) (ejm. Promedio de ventas del día de los productos femeninos)
- ✓ Operaciones lógicas Y (cuando el problema requiere que las dos condiciones se cumplan), O (cuando el problema requiere que una de las condiciones se cumpla)

Salidas(Meta): Es el resultado final, producto de las operaciones realizadas en el proceso, después de haber asegurado la verificación del mismo.

Es importante tener en cuenta que la solución de un problema implica el uso de diferentes tipos de estructuras de acuerdo a su naturaleza.

VARIABLES, CONSTANTES Y OPERADORES

Variables: Es aquel espacio en memoria que definimos y que a lo largo de la ejecución la solución puede cambiar su valor almacenado de acuerdo a las operaciones indicadas.

¿Cómo se le asigna un dato a una variable? utilizando el signo ($=$). En programación este signo se usa para ejecutar las operaciones que están a la derecha del mismo y almacenar los resultados en la variable que se encuentra a la izquierda.

Ejemplo

$a=7$ en la variable “a” se almacena el valor 7

$b=3$ en la variable “b” se almacena el valor

$a=a+b$ en este caso la variable “a” cambia de valor de acuerdo a unas operaciones indicadas. En el ejemplo “a” valía 7 y “b” 3, ahora “a” va a contener el resultado de la suma de $7+3=10$

$a=10$

Acumulador: En el caso anterior necesitamos que en una sola variable (a) quedara almacenado el total de la operación, teniendo en cuenta que el acumulador siempre debe iniciar en cero, ejemplo:

$ac=0$ (acumulador)

$b=5$

$ac=ac+b$

Contador: se utiliza cuando necesitamos saber el número de veces que se realiza una acción, a la variable se le asigna el valor actual de la variable más 1 ejm: $a=a+1$

Constantes: Es aquel espacio en memoria que definimos y que a lo largo de la ejecución de la solución no cambia, es decir luego de inicializarse permanecerá con ese valor hasta el final de la ejecución del programa.

Ejemplo

Una empresa X realiza un descuento del 10% por la compra superiores a \$400.000

$Descuento=0.1;$



En este caso como es un valor asignado por la empresa no podrá ser modificado en el programa, hasta que la empresa decida cambiarlo o suprimirlo.

Operadores

Los operadores son signos que nos permiten expresar relaciones entre variables y/o constantes, relaciones de las cuales normalmente se desprende un resultado, en un algoritmo computacional se utilizan los siguientes operadores:

Aritmético:	Operador	Significado
	+	Suma
	-	Resta
	*	Multiplicación
	/	División
Relacional:	^	Exponencial
	Operador	Significado
	=	Significado
	>	Mayor que
	<	Menor que
	<>	No igual
Lógico	>=	Mayor o igual que
	<=	Menor o igual que
	Operador	Significado
	not	no lógico.
	and	y lógico.
	or	o lógico.

EJEMPLO DE ALGORITMO

Una tienda necesita un sistema que calcule la devolución que se le debe entregar al cliente, suponiendo que cancela con un billete de mayor valor a la compra. Para que el sistema funcione se requiere: tener disponible dinero en efectivo y el valor del producto.

Entradas:	Valor de la compra y valor del billete
Procesos	Restar al valor del billete el valor de la compra
Salidas	Devolución

¿Cómo se diseña un algoritmo?

Ejemplo:

Inicio

1. Solicitar el valor de la compra
2. Solicitar el valor del billete
3. Restar al valor del billete al valor de la compra
4. Mostrar el resultado de la resta

Fin

C. EJERCITACIÓN

TRABAJO EN PAREJAS

Diseñar un algoritmo para determinar la nota que obtendrá un alumno considerando que realiza tres exámenes, de los cuales el primero y el segundo tienen una ponderación de 25%, mientras que el tercero de 50%. Se debe tener en cuenta que las notas de la universidad van de 0 a 5

Tip

- El primer carácter debe ser una letra
- Puede contener letras, números y algunos símbolos especiales (solo después del primer carácter)
- No permite espacios el blanco, iniciar con números o símbolos especiales, ni utilizar tildes, ni ñ
- Para expresar un valor en porcentaje se debe convertir a número ($4\% = 0,04$)

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor

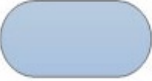






¿COMO SE REPRESENTAN LOS ALGORITMOS?

Dentro de las herramientas más usadas para la representación de los algoritmos están: el diagrama de flujo y el Pseudocódigo

¿Qué es un diagrama de flujo?

Es una representación gráfica de un algoritmo, en el cual se usa un grupo de símbolos que se conocen en todo el mundo y que son utilizados por organismos como ANSI o American National Institute e ISO (Internacional Standard Organization).

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

¿Cómo hacer un diagrama de flujo? Diagramas de Flujo (SIMBOLOGÍA Y CONSTRUCCIÓN) ([video](#)³)

Ejercicio 1 - ANÁLISIS, CONSTRUCCIÓN Y PRUEBA DE ESCRITORIO. [Ejemplo](#)⁴

C. EJERCITACIÓN

TRABAJO EN PAREJAS

Continuando con el ejercicio propuesto anteriormente, represéntelo en un diagrama de flujo: Determinar la nota que obtendrá un alumno considerando que realiza tres exámenes, de los cuales el primero y el segundo tienen una ponderación de 25%, mientras que el tercero de 50%. (Se debe tener en cuenta que las notas de la universidad van de 0 a 5)

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor.

³ Pasos por ingeniería (2016). Diagramas de Flujo (SIMBOLOGÍA Y CONSTRUCCIÓN). <https://www.youtube.com/watch?v=qDttSc3RQBC>. Consultado el 2 de agosto de 2019

⁴ Pasos por ingeniería (2017). DIAGRAMA DE FLUJO - Ejercicio 1 - ANÁLISIS, CONSTRUCCIÓN Y PRUEBA DE ESCRITORIO. Disponible en: <https://www.youtube.com/watch?v=IFixFhfS2LY>. Consultado el 2 de agosto de 2019

¿COMO CONSTRUIR UN PSEUDOCODIGO?

¿Qué es pseudocódigo?

El pseudocódigo es un bosquejo de código escrito que utiliza las convenciones estructurales de lenguaje de programación real que facilita la lectura humana y no requiere declaraciones de variables, código específico del sistema y algunas subrutinas propias de un lenguaje de programación. Una vez se hayan realizado las pruebas pertinentes se puede transcribir en un [lenguaje de programación](#). Muchos programadores lo usan para planificar la función de un algoritmo antes de darse a la tarea más técnica de codificarlo en un lenguaje. Es útil como herramienta y guía informal para pensar detenidamente en los procesos que debe ejecutar el programa.

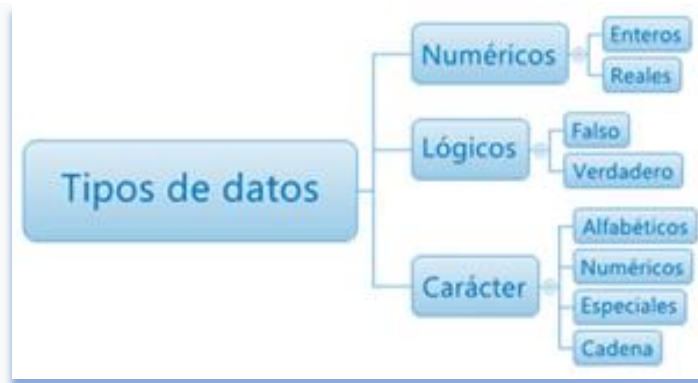
Tips para escribir un pseudocódigo

Siempre se le ha de colocar un nombre al algoritmo que describa su finalidad, de manera que sea lo primero que se lea.

¿por qué es útil el pseudocódigo?

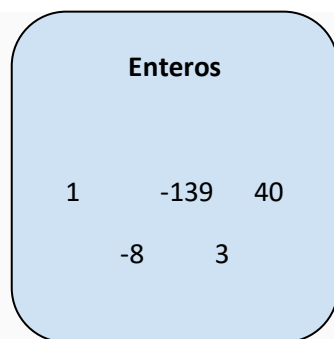
1. Se utiliza para mostrar cómo debe funcionar un algoritmo de computación
2. Sirve para explicar un proceso de computación a usuarios con menos conocimientos técnicos
3. Diseñar código en un entorno grupal de desarrollo donde se aclaran instrucciones complejas que no son comprensibles con los diagramas.

Los lenguajes de programación utilizan tipos de datos: numéricos, lógicos y de tipo caracter.



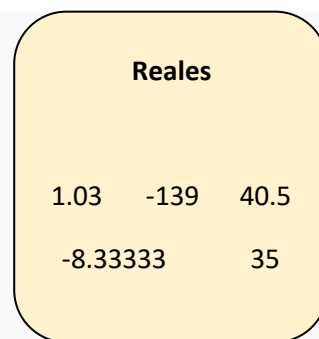
Numéricos:

Los tipos de numéricos son aquellos con los cuales se van a almacenar o van a intervenir en operaciones aritméticas o de comparación.



Nos ayudan a representar datos como:

Edad de una persona, cantidad de
elementos de un conjunto, etc



Nos ayudan a representar datos con
decimales de precisión:

valor de un objeto, temperatura.



Caracter: Son los tipos de datos que representan caracteres, letras, números con los cuales no se van a realizar operaciones aritméticas o cadenas de texto. Se escriben entre comillas simples

Caracter

"a" "hola" "34" "+" "Me gusta programar"

Lógicos: Son los tipos de datos que representan valores de tipo verdadero o falso y nos ayudan a responder preguntas cuyo resultado es verdadero o falso.

Tip

Definir en las *entradas* los tipos de datos que se utilizarán

En los *procesos* operar los valores planteados en el algoritmo

Recuerde que el pseudocódigo no sustituye el código de lenguaje de programación; solo se utiliza como un esquema de la solución del problema al resolver

Las instrucciones más utilizadas para crear un pseudocódigo son:

- **definir:** el computador recibe la instrucción de reservar un espacio en memoria de almacenamiento.
- **escribir:** el computador recibe la instrucción de mostrar un mensaje en pantalla al usuario.



- **leer**: el computador recibe la instrucción de capturar y almacenar en memoria los valores ingresados por el usuario a través del teclado
- **Si-Entonces**: el computador recibe la instrucción de comparar una opción y realizar otras instrucciones a partir del resultado de la comparación.
- **repetir hasta** : el computador recibe la instrucción de repetir instrucciones desde un inicio hasta un fin

Ejemplo:

Continuando con el ejemplo del algoritmo vamos a elaborar el pseudocódigo:

Una tienda necesita un sistema que calcule la devuelta que se le debe entregar al cliente, suponiendo que cancela con un billete de mayor valor a la compra. Para que el sistema funcione se requiere: tener disponible dinero en efectivo y el valor del producto.

Entradas:

- **Analizar el problema:** calcule la devuelta que se le debe entregar al cliente
- **Recursos:** valor de la compra, valor del billete

Salidas:

Mensaje con el resultado de la resta

Proceso:

Inicio

Escribir “Digite el valor de la compra

Leer valor_compra

Escribir “Digite el valor del billete

Leer valor_billete



UNIVERSIDAD DE
MANIZALES



Resultado = valor_billete - valor_compra

Escribir “su devuelta es:”, Resultado

Fin

C. EJERCITACIÓN

TRABAJO EN PAREJAS

Continuando con el ejercicio anterior, escríbalo en pseudocódigo: Determinar la nota que obtendrá un alumno considerando que realiza tres exámenes, de los cuales el primero y el segundo tienen una ponderación de 25%, mientras que el tercero de 50%. Se debe tener en cuenta que las notas de la universidad van de 0 a 5 (utilice si entonces)

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor

¿CÓMO ESCRIBIR PSEUDOCODIGO EN PSEINT?

Para pasar un pseudocódigo a una herramienta como Pseint⁵ se debe tener clara la sintaxis propia del lenguaje,

Cómo se definen las variables en pseint (video⁶)

Ejemplo

“Hamburguesas saludables” ofrece hamburguesas sencillas, dobles y triples, las cuales tienen un costo de \$5000, \$7000 y \$10000 respectivamente. Una familia numerosa desea comprar 13 hamburguesas sencillas, 27 dobles y 14 triples, ¿Cuánto es el total a pagar, teniendo en cuenta que por esta compra se le va a realizar un descuento del 10%.

Si realizamos esto en pseint sería de la siguiente manera:

```
Algoritmo calcular_total
  Definir ham_sen, ham_dob, ham_tri, cant_sen, cant_dob, cant_tri Como Entero
  Definir descuento, total_pagar, total Como Real

  ham_sen=5000;
  ham_dob=7000;
  ham_tri=10000;
  cant_sen=13;
  cant_dob=27;
  cant_tri=14;
  descuento=0.1;

  total=(ham_sen*cant_sen)+(ham_dob*cant_dob)+(ham_tri*cant_tri);

  total_pagar= total-(total*descuento);
  Escribir "el total a pagar es: ", total_pagar;

FinAlgoritmo
```

⁵ Tecno Hobbies (2018). Cómo descargar e instalar Pseint 2019. Disponible en: https://www.youtube.com/watch?v=wWWGSHLIRQc&list=PLwOemaB2t99YhnRhGDN-izDQt1_8-LUXG&index=2

⁶



UNIVERSIDAD DE
MANIZALES



C. EJERCITACIÓN

TRABAJO EN PAREJAS

Realizar el algoritmo y el programa en Pseint que solucione el siguiente problema

Hacer un programa para una tienda online de zapatos que determine cuánto tiene que pagar un usuario por haber comprado un par de botas en \$50.000 y un par de zapatillas en \$30.000, donde las botas tienen el 10% de descuento.

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor



¿CUÁLES SON LAS ESTRUCTURAS DE PROGRAMACIÓN?

Recordemos que los procesos que hacen parte de la solución de un problema utilizan diferentes tipos de estructuras (secuenciales, condicionales y repetitivas).

Estructuras Condicionales (animación multimedia)

Las estructuras condicionales consisten en la ejecución de pasos donde se tienen diferentes alternativas de ejecución y se requiere de decisiones que determinan el proceso a seguir. Para la evaluación de las condiciones se utilizan operadores de comparación para evaluar la condición que devuelve como resultado dos valores posibles: verdadero o falso.

Pseudocódigo

Inicio

Si (condición es verdadera) entonces

ejecuta accionesA

Sino

ejecuta accionesB

Diagrama de flujo



Ejemplo Algoritmos condicionales (doble) en PseInt – Parte 1. ([video](#))



C. EJERCITACIÓN

TRABAJO EN PAREJAS

Diseñar un programa en PseInt que calcule el promedio de 3 notas parciales de un estudiante, e indicar si el estudiante aprobó o reprobó el curso, teniendo en cuenta que se aprueba con un promedio igual o superior a 3.0

D. APLICACIÓN

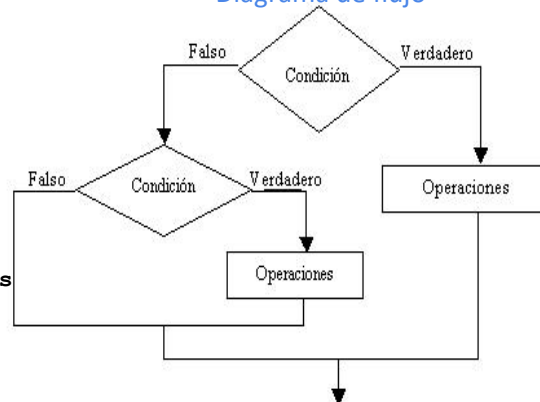
Socializamos las respuestas con los compañeros y el profesor

Condicionales anidados: Cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional, ejemplo: Algoritmos condicionales (Anidados) en PseInt - Abecedario 1a

Pseudocódigo

```
Inicio
Si (condición es verdadera) entonces
    ejecuta accionesA
Sino
    Si (condición es verdadera) entonces
        ejecuta accionesB
```

Diagrama de flujo



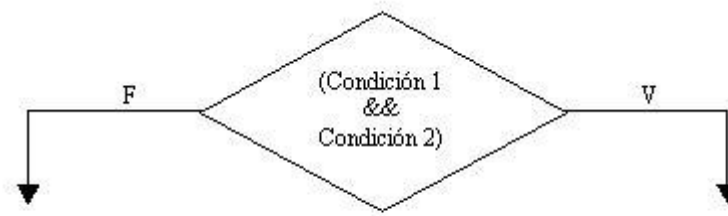


Fuente: Tutoriales Programación Ya (s.f.). Estructuras condicionales anidadas. Disponible en: <https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?punto=8&codigo=81&inicio=0>.

Condiciones compuestas con operadores lógicos

Cuando se tienen 2 o más condiciones dentro de una estructura condicional se utilizan los operadores lógicos **Y (&)**, o **(/)**

Operador &&



Traducido se lo lee como "Y". Si la Condición 1 es verdadera "Y" la condición 2 es verdadera ejecuta la rama del verdadero, si una de las dos condiciones no se cumple se ejecuta la rama del falso.

Operador II

Traducido se lo lee como "O": ejm:

Se ejecuta la rama del verdadero si:

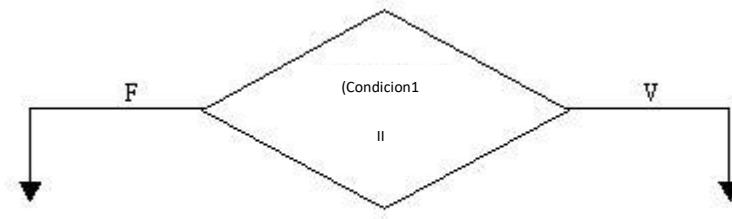
la Condición 1 es verdadera y la condición 2 es falsa

la Condición 1 es falsa y la condición 2 es verdadera

la Condición 1 es verdadera y la condición 2 es verdadera

Se ejecuta la rama del falso si

la Condición 1 es falsa y la condición 2 es falsa



Los operadores lógicos permiten plantear algoritmos más cortos y comprensibles.

Switch

Esta Instrucción, es otra manera de ejecutar la instrucción SI, cuando se requiere elegir entre más de dos opciones.

Ejemplo

Se desea diseñar un algoritmo que escriba los nombres de los días de la semana en función del valor de una variable DIA introducida por teclado. Los días de la semana son 7; por consiguiente, el rango de valores de DIA será 1..7, y caso de que DIA tome un valor fuera de este rango se deberá producir un mensaje de error advirtiendo la situación anómala.

Entradas: dia tipo entero

Salida: Mensaje indicando el dia de la semana

Pseudocódigo

Inicio

Definir dia como entero

Escribir "Digite el día:"

Leer dia



En caso de DIA sea

- 1: escribir('Lunes')
- 2: escribir('Martes')
- 3: escribir('Miércoles')
- 4: escribir('Jueves')
- 5: escribir('Viernes')
- 6: escribir('Sábado')
- 7: escribir('Domingo')

De lo contrario

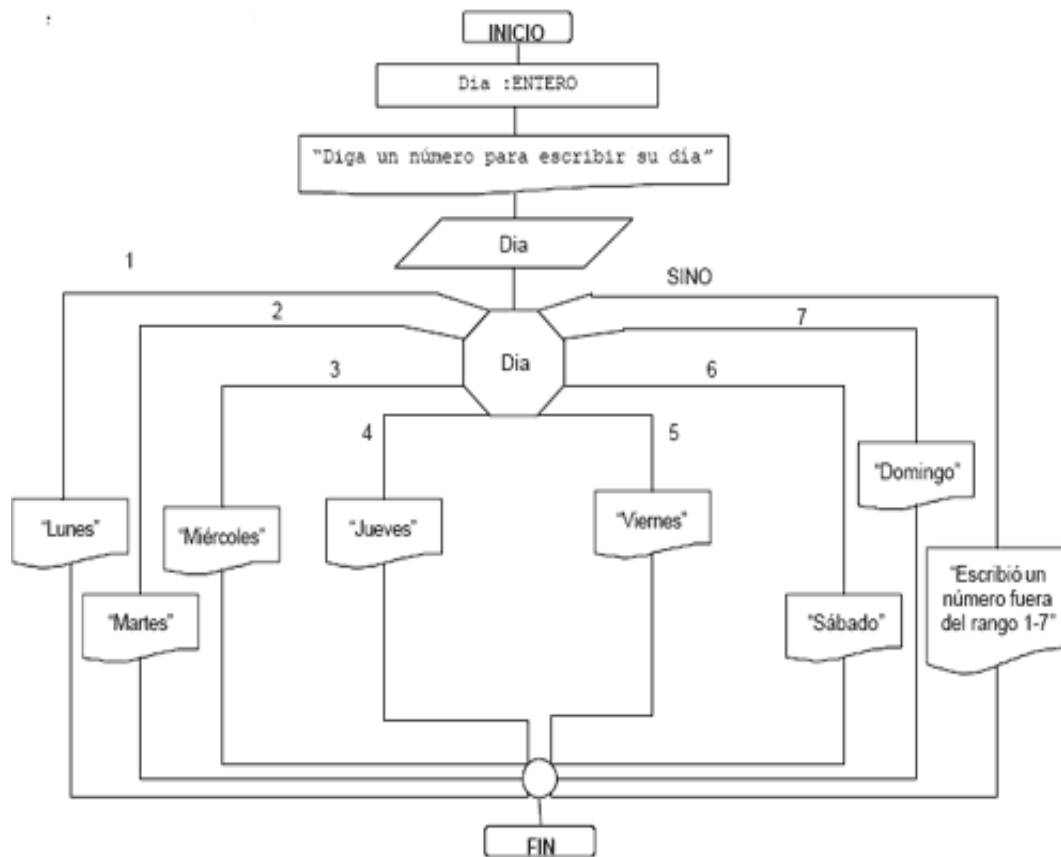
escribir('Error')

fin_según

fin Algoritmo



DIAGRAMA DE FLUJO





UNIVERSIDAD DE
MANIZALES



Programa en pseint

```
<sin_titulo>*
1  Algoritmo programa_diaSemana
2  Definir dia como entero;
3
4  Escribir "Digite un numero de 1-7 para indicarle el dia :";
5  Leer dia;
6
7  Segun dia Hacer
8  1: Escribir "Lunes";
9
10 2: Escribir "Martes";
11
12 3: Escribir "Miércoles";
13
14 4: Escribir "Jueves";
15
16 5: Escribir "Viernes";
17
18 6: Escribir "Sábado";
19
20 7: Escribir "Domingo";
21
22 De Otro Modo:
23 Escribir "Digitó un numero fuera del rango 1-7";
24
25 Fin Segun
26 FinAlgoritmo
```

PSeInt - Ejecutando proceso PROGRAMA_DI...

Es seguro | <https://www.rollapp.com/client?sessionId=ee0...>

*** Ejecución Iniciada. ***

Digite un numero de 1-7 para indicarle el dia :
> 7

Domingo

*** Ejecución Finalizada. ***

☐ No cerrar esta ventana ☐ Siempre visib

C. EJERCITACIÓN

TRABAJO EN PAREJAS

Una agencia de seguros para automóviles asigna pólizas de acuerdo al género y edad del conductor. Los varones menores de 25 años pagan \$2.000.000 y los que tengan 25 años o más pagan \$1.800.000; las mujeres de 21 años o menos pagan \$2.200.000 y las mayores de 21 años, pagan \$1.700.000. Diseñe un programa que imprima la edad del conductor, el género y el valor correspondiente a la póliza que debe pagar.

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor



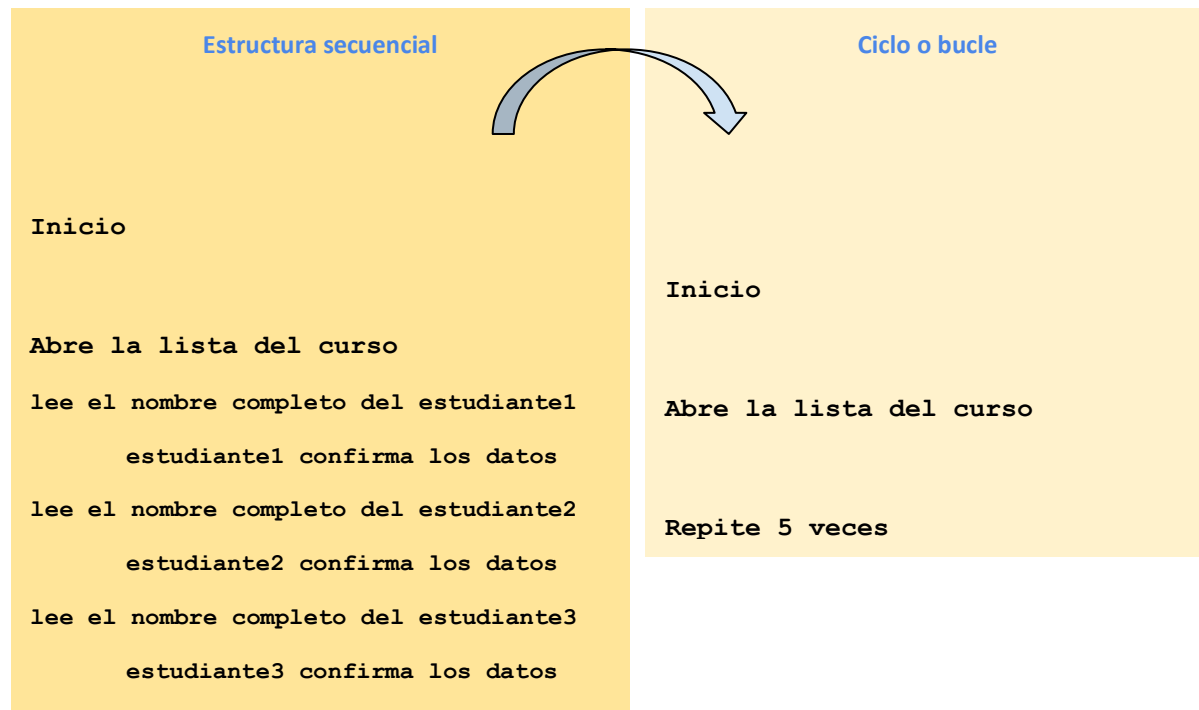
Estructura Iterativas

Una de las características importantes que se pueden aprovechar de las computadoras es precisamente su capacidad de repetir una misma instrucción a gran velocidad y con alto grado de confiabilidad.

El objetivo de las estructuras iterativas es realizar la repetición de una secuencia de instrucciones mientras se cumpla una condición. En cada caso se ejecuta un bloque de instrucciones mientras la condición que se evalúa tome valor verdadero.

Su lógica de funcionamiento sería la siguiente:

La profesora de informática necesita confirmar los datos de 5 estudiantes, el algoritmo inicial sería:



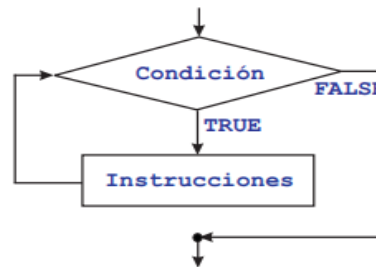
En el caso anterior, se evidencia que es mucho más eficiente y ágil utilizar las estructuras de iteración en la solución de problemas que involucren tareas repetitivas.



Ciclo while (mientras)

Permite repetir instrucciones cuando la condición evaluada sea verdadera y en el momento en que la condición sea falsa saldrá del ciclo y continuará con la siguiente instrucción, si es el caso. En el while la condición se evalúa al inicio del ciclo si la primera vez que se evalúa la condición es falsa, el ciclo no se ejecuta.

```
...  
while expresión-lógica  
    instrucciones  
end  
...
```



Ejemplo ([video](#)):

C.EJERCITACIÓN

TRABAJO EN PAREJAS

Vamos a realizar un juego para adivinar un número generado aleatoriamente del 1 al 100, y vamos a tener 10 oportunidades de acertar. El juego debe indicar si el número aleatorio elegido por el jugador está por encima del digitado o por debajo, en caso de ser iguales debe notificar al jugador que ha adivinado...!!!

NOTA: Para generar el aleatorio en Pseint, se utiliza la función `azar()`

D. APLICACIÓN

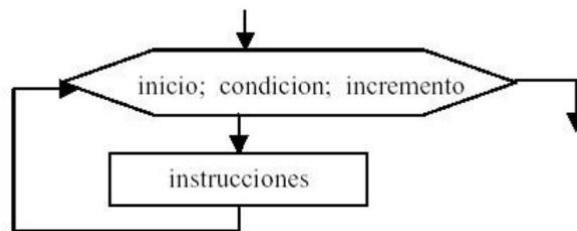
Socializamos las respuestas con los compañeros y el profesor



Ciclo for (para)

Las estructuras **for** nos permiten ejecutar una o varias instrucciones conociendo el valor inicial y el valor final (condición), además determinar el rango de incremento del valor inicial.

La representación del diagrama de flujo para un ciclo FOR (Para) es:



Un ciclo **PARA** se compone de:

Variable: (variable numérica) que indica el valor inicial

Hasta: donde finaliza (condición)

Paso: el rango

Ejemplo:

```
Algoritmo ciclo_Para
  Para variable_numerica<-valor_inicial Hasta valor_final Con Paso paso Hacer
    ...
    secuencia_de_acciones
  Fin Para
FinAlgoritmo
```

Algoritmo ciclo_para

Para i<-1 Hasta 5 Con Paso 1 Hacer

Escribir i

Fin Para

FinAlgoritmo



¿Cómo funciona el FOR?

- ✓ El valor inicial de la variable solo se ejecuta una vez al inicio del ciclo.
 - ✓ Se ejecuta la condición, mientras la variable inicial no supere el valor final se ejecuta las instrucciones
 - ✓ Se ejecuta el incremento o decremento de la variable.
 - ✓ Pasa a preguntar por la condición.
-
- ✓ Si la condición fuera falsa la ejecución pasará a la siguiente instrucción.

Ejemplo ([video](#)):

C. EJERCITACIÓN

TRABAJO EN PAREJAS

Se requiere un programa para saber cuánto ha ahorrado una persona durante un semestre, teniendo en cuenta que al final de cada mes guarda diferentes cantidades de dinero; el programa debe calcular el ahorro parcial (mes a mes) y el valor total ahorrado durante el semestre. Se debe validar que los valores sean mayores a 0.

D. APLICACIÓN

Socializamos las respuestas con los compañeros y el profesor



EN LA WEB TAMBIÉN SE APRENDE

Baeza J. (2015). Estructuras Iterativas. Disponible en: <https://www.youtube.com/watch?v=Hq0Bzb6bsi8>.

Ecured.cu (s.f.). Diagramas de flujo. Disponible en: https://www.ecured.cu/Diagrama_de_Flujo.

_____ (s.f.). Pseudocódigo. Disponible en: <https://www.ecured.cu/Pseudoc%C3%B3digo>.

Hektor Profe (2014). Curso Programación [#11] Tipos de operadores. Disponible en: https://www.youtube.com/watch?v=P_cpMPfto_Q.

Magic Markers (2015). ¿Qué es un algoritmo?. Disponible en: <https://www.youtube.com/watch?v=U3CGMyjzlvM>.

Micro TICS (2017). Algoritmo secuencial en PSeInt - No1. Disponible en: <https://www.youtube.com/watch?v=QfjVuXL8ih0&list=PLvmfsA3Ty0tmTMy9cqgbf69A4UdfrFnzp&index=2>.

_____ (2017). Algoritmo condicional simple en PSeInt - No 1. Disponible en: <https://www.youtube.com/watch?v=vhIJMZ0b2E8&list=PLvmfsA3Ty0tmTMy9cqgbf69A4UdfrFnzp&index=13>.

_____ (2017). Algoritmos condicionales (doble) en PseInt - Parte 1 de 6. Disponible en: <https://www.youtube.com/watch?v=af32A8hSUHg&list=PLvmfsA3Ty0tmTMy9cqgbf69A4UdfrFnzp&index=20>.

_____ (2018). Algoritmos condicionales (Anidados) en PseInt - Abecedario1a. Disponible en: <https://www.youtube.com/watch?v=94rCixEbEIM&list=PLvmfsA3Ty0tmTMy9cqgbf69A4UdfrFnzp&index=23>.

Pasos por ingeniería (2016). ¿Qué es un algoritmo? (definición, características, tipos, operadores y variables). Disponible en: <https://www.youtube.com/watch?v=Tu9OQSff-gw>.

_____ (2016). ¿Cómo hacer un diagrama de flujo? Disponible en:
<https://www.youtube.com/watch?v=qDttSc3RQBc>.

Programación ATS (2017). 8. Programación desde Cero | Entidades Primitivas | Identificadores (Constantes y Variables). Disponible en: <https://www.youtube.com/watch?v=ZOoOFyR2QVI>

Sandoval B. (2014). Algoritmos - Variables en PseInt. Disponible en:
<https://www.youtube.com/watch?v=SfLDY5EzmDI>

Tecno Hobbies (2018). Pseint 2018 | Ciclos | Ciclo Para en Pseint. Disponible en:
https://www.youtube.com/watch?v=UF4NUilOB9M&list=PLwOemaB2t99YhnRhGDN-izDQt1_8-LUXG&index=13.

_____. (2018). Pseint 2019 | Ciclos | como usar mientras en Pseint + 2 ejercicios. Disponible en:
https://www.youtube.com/watch?v=Mabj-CYJ6GQ&list=PLwOemaB2t99YhnRhGDN-izDQt1_8-LUXG&index=14

_____ (2019). ¿Cómo descargar e instalar pseint 2019?. Disponible en:
<https://www.youtube.com/watch?v=wWWGSHLIRQc>.

Vanegas B. F (2013). ¿Porqué es importante aprender a programar?. Disponible en:
<https://www.youtube.com/watch?v=yCn0atPy3e4>.