



# Universidad Autónoma del Estado de México

## Facultad de Medicina

Licenciatura en Bioingeniería Médica

Unidad de Aprendizaje:  
Algoritmos y programación básica

Unidad 3:  
Estructuras de control de flujo en pseudocódigos

Tema: Estructuras secuenciales y de selección

***Presenta: Dr. Jorge Rodríguez Arce.***





## Unidad 2. Tipos, operadores y expresiones

**Objetivo:** Identificar los tipos, operadores y expresiones utilizados en computación para escribir expresiones sintácticamente correctas utilizadas en la resolución de problemas.

- Tipos de datos simples
- Identificadores
- Constantes
- Operadores (aritméticos, lógicos, relacionales)
- Orden de precedencia y evaluación

## Unidad 3. Estructuras de control en diagrama de Flujo y pseudocódigo

**Objetivo:** Aplicar las estructuras de control computacionales utilizando instrucciones sintáctica y semánticamente correctas para controlar el flujo de control en un programa computacional expresado en diagrama de flujo y/o pseudocódigo, utilizar la técnica de pruebas de escritorio para validar la propuesta de solución

- Secuenciales (entrada, salida, asignación)
- Selección (simples, dobles, múltiples y anidadas)
- Iteración o Repetitivas (mientras, repite, para y anidadas)

## Unidad 4. Introducción a la Programación Modular

**Objetivo:** Utilizar el paradigma de la programación modular para resolver problemas complejos identificando módulos que permiten resolver el problema en subproblemas más simples o sencillos.

- Programación Modular (Módulo Principal y submódulos)
- Procedimientos y Funciones
- Variables Globales y Locales
- Paso de parámetros





# Introducción

La unidad de aprendizaje de Algoritmos y Programación Básica, esta integrada al núcleo de formación integral de la Lic. en Bioingeniería Médica, porque los alumnos requieren adquirir las competencias necesarias en la implementación de algoritmos y en el uso del pseudocódigo. Dichas competencias les servirán para implementar soluciones de algoritmos en distintos sistemas por ejemplo: escribiendo el software para un sistema computacional típico, para diseñar e implementar algoritmos de análisis de imágenes médicas o escribiendo el software para otras plataformas especializadas como procesadores digitales de señales o microcontroladores.





# Objetivo

- El presente material tiene como objetivo presentar dos de las tres estructuras usadas para controlar el flujo de control en un programa expresado en pseudocódigo.
- Las estructuras estudiadas en este material se dividen en secuenciales y de selección.
- El alumno será capaz de emplear las estructuras secuenciales y de selección para escribir algoritmos de programas en los que se involucre el cálculo de datos y la toma de decisiones.





# Índice de contenidos

- **Operaciones básicas en algoritmos**
  - Operaciones de asignación, lectura y escritura
- **Estructuras secuenciales**
  - Entrada
  - Salida
  - Asignación
- **Estructuras de selección**
  - Estructura de selección simple
  - Estructura de selección doble
  - Estructura de selección anidada
  - Estructura de selección múltiple





# ¿Qué son las estructuras de flujo de control de un programa?

- En un algoritmo a menos que se especifique algo distinto el orden de evaluación de cada instrucción es siempre **secuencial**, lo que significa que las sentencias se ejecutan una detrás de otra en el orden en que están escritas.
- Las estructuras de **flujo de control** permiten que dicho flujo secuencial sea modificado de un modo preciso.
- Las **estructuras de flujo** se dividen en :
  - **Estructuras de selección:** se utilizan para decidir en base a una condición el conjunto de sentencias que tienen que ser ejecutadas
  - **Estructuras de repetición:** se utilizan para repetir la ejecución de un conjunto de sentencias.





# Operaciones básicas en algoritmos





# Instrucciones de asignación

- Sirve para asignar un valor a una variable.
- Se simboliza con el símbolo “ $\leftarrow$ ” para evitar confusiones con el operador de igualdad (=).
- Ejemplos:
  - $a \leftarrow 5$ : asigna a la variable “a” el valor de 5
  - $x \leftarrow y+2$ : asigna a la variable “x” el resultado de la suma  $y+2$







# Instrucción de entrada: Leer

- Se emplea para indicar que se debe de leer uno o varios datos de entrada, indicando en una lista cada una de las variables en las que se almacena el valor leído.
- Es importante que el dato leído sea del mismo tipo de la variable en la que se va a almacenar.

–Ejemplos:

leer (edad) : indica que se lee un dato de entrada y se guarda en la variable edad

leer (horas, minutos, segundos) : indica que se leen tres datos de entrada, el primer dato se guarda en la variable horas, el segundo dato en la variable minutos y el tercero en la variable segundos.





# Instrucción de salida: Escribir

- Se emplea para mostrar alguna información en pantalla al usuario. La información que se puede mostrar en pantalla es:
    - Un mensaje al usuario
    - El contenido de una variable
    - Una combinación de las dos anteriores
- Ejemplos:
1. Escribir (“Hola me llamo Juan”): mostrara en pantalla el mensaje que esta entre comillas.
  2. Escribir (x): mostrará en pantalla el contenido de la variable x
  3. Escribir (“El total de la suma es: ” + total): mostrará en pantalla el mensaje entre comillas junto con el valor de la variable total.





# Estructuras secuenciales





# Estructura secuencial

- La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las acciones se ejecutan de tal modo que al finalizar la instrucción 1 se continua con la 2 después con la 3 y así sucesivamente hasta el final del algoritmo.
- Un algoritmo con estructura secuencial utiliza solo las operaciones básicas: asignación, escritura y lectura.

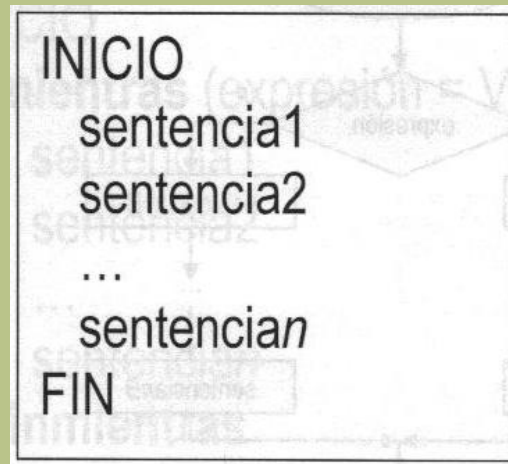


Imagen tomada de: Joyanes, A. L., (2008), *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*, Madrid, McGraw-Hill.





# Ejemplo

## Estructuras secuenciales

- Escriba el algoritmo de un programa que calcule la suma y producto de dos números enteros.

Variables:

Entero: a, b, s, p

Inicio

Escribir (“Programa que calcula la suma de dos números”)

Escribir (“Escribe el valor del primer número”)

Leer (a)

Escribir (“Escribe el valor del segundo número”)

Leer (b)

$s \leftarrow a+b$

$p \leftarrow a*b$

Escribir (“El resultado de la suma es :” + s)

Escribir (“El resultado del producto es: “ + p)

Fin





# Ejercicios de tarea

## Estructuras secuenciales

- Instrucciones: escribir un algoritmo que resuelva cada uno de los siguientes problemas.
  - 1) Calcular el índice de masa corporal de un paciente .
  - 2) Leer el peso de un hombre en libras y mostrar su peso en kilogramos (nota: una libra equivale a 0.453592 kilogramos).
  - 3) Hacer un programa que calcule la paga neta de un médico conociendo el número de horas trabajadas, la tarifa por hora y el porcentaje de impuestos.





# Estructuras de selección





# Estructuras de selección

- Las estructuras selectivas se emplean para evaluar una condición y en función del resultado de la misma se realiza un conjunto de instrucciones específico, de esta manera la ejecución del algoritmo no es completamente secuencial.
- La condición puede ser una sola o un conjunto de condiciones evaluadas empleando los operadores lógicos “y” y “o”.
- La condición o condiciones se especifican usando expresiones lógicas, las cuales al evaluarse solo podrán ser verdades (si se cumple la condición) o falsas (no se cumple la condición).







# Estructuras de selección

- Las estructuras selectivas pueden ser:
  - Simples
    - Si la condición es verdadera se ejecuta un conjunto de instrucciones en otro caso no se realiza nada.
  - Dobles
    - Si la condición es verdadera se ejecuta un conjunto de instrucciones en otro caso se ejecuta un conjunto de instrucciones diferente.
  - Múltiples
    - Es cuando la condición puede tener valores distintos a solo verdadero o falso.





# Estructura de selección simple

- La **estructura de selección simple** al evaluar la condición sucede que:
  - sí la condición es verdadera entonces ejecuta el conjunto de instrucciones indicadas,
  - sí la condición es falsa entonces no se ejecuta ninguna instrucción.

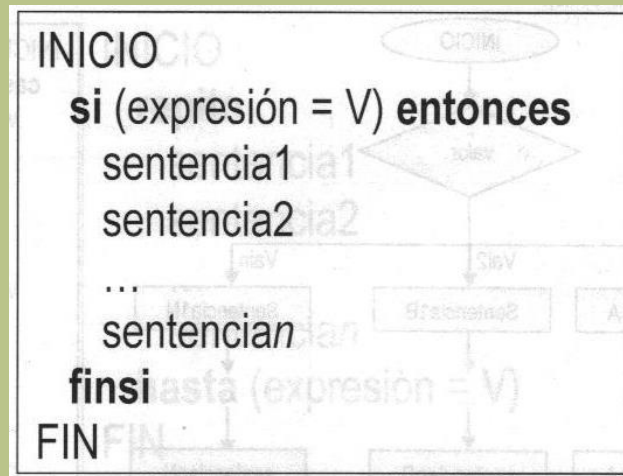


Imagen tomada de: Joyanes, A. L., (2008), *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*, Madrid, McGraw-Hill.





# Ejemplo de selección simple

- Escribe un algoritmo que realice la suma de dos números enteros solo si son positivos.

Variables:

Entero: n1, n2, s

Inicio

Escribir (“Programa que calcula la suma de dos números positivos”)

Escribir (“Escribe el valor del primer número”)

Leer (n1)

Escribir (“Escribe el valor del segundo número”)

Leer (n2)

**Si (n1 > -1 y n2 > -1) entonces**

**s ← a+b**

**Escribir (“El resultado de la suma es :” + s)**

**fin si**

Fin





# Ejercicios de tarea de selección simple

- Instrucciones: escribir un algoritmo que resuelva cada uno de los siguientes problemas.
- 1) Realizar un algoritmo que pregunta por las 3 calificaciones parciales de un alumno, calcula el promedio y se lo muestra al usuario, además si el promedio es mayor o igual a 6.0 muestra un mensaje para felicitarlo porque acredita el curso.
  - 2) Para calcular el precio de entrada en un parque de atracciones a las personas se les pregunta su año de nacimiento. De esta manera el programa calcula y muestra la edad de la persona. En caso de que la edad sea menor a 15 años se muestra un mensaje informando que se tiene 50% de descuento en el precio de entrada.





# Estructura de selección doble

- La alternativa doble **permite elegir entre dos alternativas posibles**, en función del cumplimiento o no de una determinada condición. Es decir, si la condición es verdadera, entonces se ejecuta un conjunto de instrucciones y si es falsa, se ejecuta otro conjunto de instrucciones.

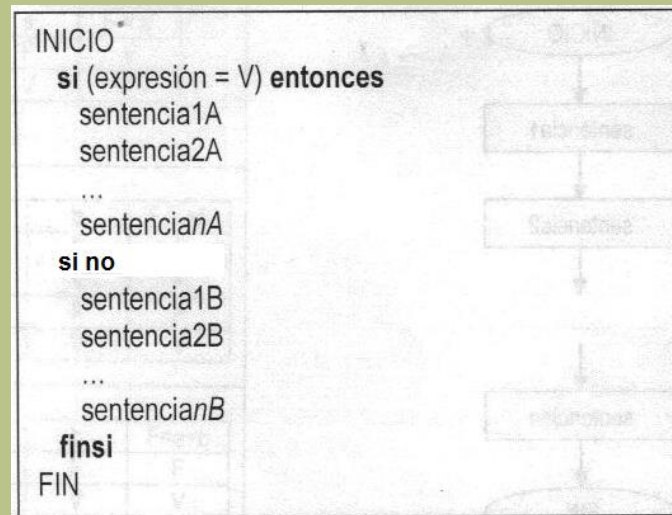


Imagen tomada de: Joyanes, A. L., (2008), *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*, Madrid, McGraw-Hill.





# Ejemplo de selección doble

- Escribe un algoritmo que realice la suma de dos números enteros solo si son positivos, en caso de que los números no sean positivos mostrar un mensaje de error al usuario.

Variables:

Entero: n1, n2, s

Inicio

Escribir (“Programa que calcula la suma de dos números positivos”)

Escribir (“Escribe el valor del primer número”)

Leer (n1)

Escribir (“Escribe el valor del segundo número”)

Leer (n2)

**Si (n1 > -1 y n2 > -1) entonces**

**s ← a+b**

**Escribir (“El resultado de la suma es :” + s)**

**Si no**

**Escribir (“Error: los números tienen que ser positivos”)**

**fin si**

Fin





# Ejercicios de tarea de selección doble

- Instrucciones: escribir un algoritmo que resuelva cada uno de los siguientes problemas.
- 1) Realizar un programa que pida una clave de acceso al usuario. Si la clave de acceso es correcta preguntar por dos números al usuario y calcular la suma. Si la clave es incorrecta mostrar un mensaje de error y terminar el programa.
  - 2) Preguntar por su calificación al usuario, si la calificación es mayor o igual a 60 felicitarlo porque acredita el curso, además si la calificación es mayor a 90 decirle que es un excelente alumno. En caso de que su calificación sea menor a 60 decirle que tendrá que repetir el curso.





# Estructuras anidadas

- Una estructura si-entonces puede contener otra estructura si-entonces y esta estructura si-entonces puede contener otras y así sucesivamente cualquier número de veces; a su vez, dentro de cada estructura pueden existir diferentes acciones.

```
si <condicion1> entonces
  si <condicion2> entonces
    .
    .
    .
    <acciones>
  fin_si
fin_si
```

Imagen tomada de: Joyanes, A. L., (2008), *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*, Madrid, McGraw-Hill.







# Ejemplo de estructuras anidadas

- Escriba un algoritmo que lea tres números  $n1$ ,  $n2$  y  $n3$  y muestre en pantalla el valor más grande. Se supone que los tres números son diferentes.

Variables:

Entero:  $n1$ ,  $n2$ ,  $n3$ , mayor

Inicio

Leer ( $n1$ ,  $n2$ ,  $n3$ )

Si  $n1 > n2$  entonces

    Si  $n1 > n3$  entonces

        mayor  $\leftarrow n1$

    Si no

        mayor  $\leftarrow n3$

    Fin\_si

Si no

    Si  $n2 > n3$  entonces

        mayor  $\leftarrow n2$

    Si no

        mayor  $\leftarrow n3$

    Fin\_si

Escribir ("El valor mas grande es: " + mayor)

Fin





# Ejercicios de tarea de selección anidada

- Instrucciones: escribir un algoritmo que resuelva cada uno de los siguientes problemas.
- 1. Escriba un algoritmo que lee tres número diferentes e imprime los valores máximo y mínimo
- 2. Escribir un algoritmo que permita calcular las soluciones de una ecuación de segundo grado, incluyendo los valores imaginarios





# Estructura de selección múltiple

- Sirve para evaluar una expresión que podrá tomar  $n$  valores distintos, es decir existen más de dos elecciones posibles.
- Según se elija uno de los posibles valores en la condición, se realizará una de las  $n$  acciones, por lo que el flujo del algoritmo seguirá un determinado camino entre los  $n$  posibles valores.
- Se debe especificar cada uno de los valores que puede tomar la expresión, por lo que su uso está prohibido para valores decimales y cadenas de texto.

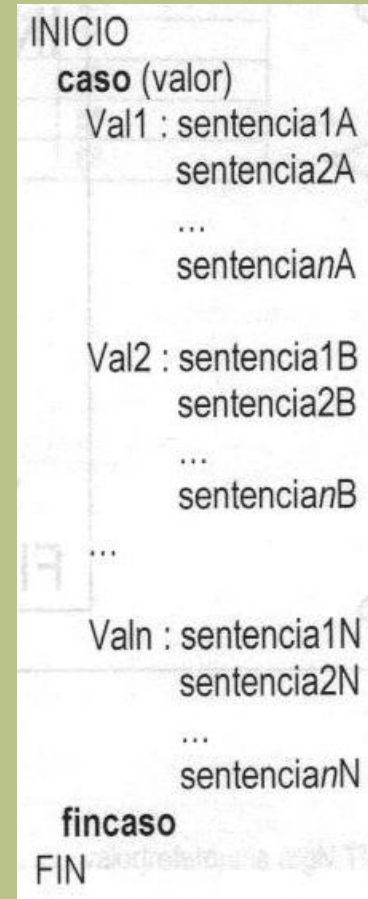


Imagen tomada de: Joyanes, A. L., (2008), *Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos*, Madrid, McGraw-Hill.





# Ejemplo de selección múltiple

- Escriba un algoritmo que escriba los nombres de los días de la semana en función del valor de una variable numérica de entrada. Por ejemplo si el usuario escribe 5 el programa muestra el nombre de Viernes. Si el usuario escribe un número fuera del rango de 1 a 7 el programa muestra un mensaje de error.

Variables:

Entero: valor

Inicio

Leer (valor)

Caso (valor) hacer

1: Escribir (“El día es Lunes”)

2: Escribir (“El día es Martes”)

3: Escribir (“El día es Miércoles”)

4: Escribir (“El día es Jueves”)

5: Escribir (“El día es Viernes”)

6: Escribir (“El día es Sábado”)

7: Escribir (“El día es Domingo”)

Otro: Escribir (“Error el valor no es incorrecto”)

Fin casi

Fin





# Ejercicios de tarea de selección múltiple

- Instrucciones: escribir un algoritmo que resuelva cada uno de los siguientes problemas.
1. Hacer un programa que pide al usuario un número del 1 al 12 que representan el número de mes del año y como salida imprime: el nombre del mes y la cantidad de días que tiene dicho mes.
    - a) Considera que febrero tiene 28 días
    - b) Recuerda que abril, junio, septiembre y noviembre tienen 30 días.





# Ejercicios de tarea de selección múltiple

- 2. Escribir un programa que pida al usuario un número de día y un número de mes, evalúe si la entrada es correcta y en caso de que así sea calcule el día del año que es, considerando un año no bisiesto.

Ejemplo 1:

- Día: 15
- Mes: 2
- El ordenador muestra: El día 15 del mes 2 es el día 46 del año.

Ejemplo 2:

- Día: 40
- Mes: 2
- El ordenador muestra: Datos incorrectos





# Resumen





# Resumen

- El flujo normal de control de un algoritmo es secuencial, es decir se ejecuta instrucción por instrucción en orden.
- Las estructuras de control de selección y repetición alteran el flujo de ejecución secuencial del algoritmo
- Las estructuras de selección sirven para determinar cuáles sentencias se han de ejecutar a continuación en base a si una cierta condición al ser evaluada es cierta o falsa.
- Las estructuras de selección son: si-entonces (simple), si-sino-entonces (doble) y caso (múltiple)







# Bibliografía

## BASICA

- Norton, P., (2006), ***Introducción a la computación***, México, McGraw-Hill.
- Joyanes, A. L., (2008), ***Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos***, Madrid, McGraw-Hill.
- Rodríguez, B. L., Fernández A. M. y Joyanes A. L., (2003), ***Fundamentos de programación. Libro de problemas***. España, McGrawHill.





# Bibliografía

## COMPLEMENTARIA

- Cairo, O., (2008), ***Metodología de la programación: Algoritmos, Diagramas de Flujo y programas***, México, Alfaomega.
- Ramírez, F., (2007), ***Introducción a la programación: Algoritmos y su implementación en VB.NET, C#, JAVA Y C++***, México, Alfaomega.
- López, R. L., (2004), ***Programación estructurada. Un enfoque algorítmico***, México, Alfaomega.

