



Sesión 3 – Semana 2

Estructuras de Selección

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



Contenido

- Introducción
- Estructura de selección simple
- Estructura de selección compuesta
- Estructuras de selección anidadas
- Estructura de selección múltiple

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Introducción

- Hasta el momento los algoritmos que hemos diseñado se desarrollan de manera secuencial. Todas las instrucciones presentes en el algoritmo se ejecutan en el orden especificado.
- Frecuentemente, las soluciones de los problemas requieren que se ejecute un grupo de instrucciones en lugar de otro dependiendo de alguna condición.
- Las estructuras de selección se utilizan para decidir si un grupo de instrucciones se debe ejecutar o no.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Estructura de selección simple en pseudocódigo

```
⋮  
si expresión lógica entonces  
    // Instrucciones  
fin_si  
⋮
```

Si la **expresión lógica** evalúa a **verdadero** se ejecutan las instrucciones que se encuentran entre las palabras reservadas **entonces** y **fin_si**. Si la **expresión lógica** evalúa a **falso** se ejecutan las instrucciones que se encuentren después de la palabra reservada **fin_si**.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G

Estructura de selección compuesta en pseudocódigo

```
⋮  
si expresión lógica entonces  
    // Instrucciones  
si_no  
    // Instrucciones  
fin_si  
⋮
```

Si la **expresión lógica** evalúa a **verdadero** sólo se ejecutan las instrucciones que se encuentran entre las palabras reservadas **entonces** y **si_no**. Si la **expresión lógica** evalúa a **falso** sólo se ejecutan la instrucciones que se encuentren entre las palabras reservadas **si_no** y **fin_si**.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Ejemplo 1: Estructura de selección simple

ALGORITMO

```
algoritmo RaizCuadrada
variables
  real: numero, r
inicio
  muestre('CALCULAR RAIZ CUADRADA DE
  UN NUMERO NO NEGATIVO')
  muestre('Entre un numero: ')
  lea(numero)
  si numero  $\geq$  0 entonces
    r  $\leftarrow$  raiz2(numero)
    muestre('La raíz cuadrada es: ', r)
  fin_si
fin
```

PROGRAMA

```
disp('CALCULAR RAIZ CUADRADA DE UN
  NUMERO NO NEGATIVO')
numero = input('Entre un numero: ');

if numero >= 0
  r = sqrt(numero)
  disp('La raíz cuadrada es: ')
  disp(r)
end
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Ejemplo 2: Estructura de selección compuesta

ALGORITMO

```
algoritmo RaizCuadrada2
variables
    real: numero, raiz
inicio
    muestre('CALCULAR RAIZ CUADRADA')
    muestre('Entre un numero: ')
    lea(numero)
    si numero  $\geq$  0 entonces
        r  $\leftarrow$  raiz2(numero)
        muestre('La raíz cuadrada es: ', r)
    si_no
        muestre('Error: En los reales no hay raíz cuadrada de números negativos')
    fin_si
fin
```

PROGRAMA

```
disp('CALCULAR RAIZ CUADRADA')
numero = input('Entre un numero: ');

if numero >= 0
    r = sqrt(numero)
    disp('La raíz cuadrada es: ')
    disp(r)
else
    disp('Error: En los reales no hay raíz cuadrada de números negativos')
end
```

Ejemplo 3: Estructura de selección compuesta

ALGORITMO

```
algoritmo MenorMayorDeEdad
variables
    entero: edad
inicio
    muestre('Entre su edad: ')
    lea(edad)
    si edad ≥ 18 entonces
        muestre('Usted es mayor de edad')
    si_no
        muestre('Usted es menor de edad')
    fin_si
fin
```

PROGRAMA

```
edad = input('Entre su edad: ');

if edad >= 18
    disp('Usted es mayor de edad')
else
    disp('Usted es menor de edad')
end
```




Ejemplo 4

Diseñe un algoritmo que lea las notas de un estudiante correspondientes a los dos seguimientos y al parcial del curso de Fundamentos de Programación y muestre la nota acumulada actual y la nota mínima que debe obtener en el examen final para aprobar el curso. En caso que el estudiante no necesite nota para el final muestre un mensaje que indique que el estudiante ya aprobó el curso.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Algoritmo para del ejemplo 4

algoritmo NotaEstudiante2

variables

real: seguimiento1, seguimiento2, parcial, final, notaAcumulada

inicio

muestre('CALCULAR NOTA DE ESTUDIANTE PARA EL EXAMEN FINAL')

muestre('Ingrese la nota del primer seguimiento: ')

lea(seguimiento1)

muestre('Ingrese la nota del parcial: ')

lea(parcial)

muestre('Ingrese la nota del segundo seguimiento: ')

lea(seguimiento2)

$\text{notaAcumulada} \leftarrow 0.2 * \text{seguimiento1} + 0.3 * \text{parcial} + 0.2 * \text{seguimiento2}$

muestre('La nota acumulada es: ', notaAcumulada)

si $\text{notaAcumulada} \geq 3.0$ entonces

muestre('Ya gano el curso. Felicitaciones!')

si_no

$\text{final} \leftarrow (3.0 - \text{notaAcumulada}) / 0.3$

muestre('En el final debe obtener una nota de ', final, ' para ganar el curso.')

fin_si

fin

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Estructuras de selección anidadas

- La estructura de selección simple permite determinar si una secuencia de instrucciones se debe ejecutar. La estructura de selección compuesta permite determinar cuál secuencia de instrucciones, de dos posibles, se deben ejecutar.
- Las estructuras de selección anidadas permiten elegir, de varias secuencias de instrucciones posibles, cuál secuencia se debe ejecutar.
- Las estructuras de selección anidadas se hacen escribiendo estructuras de selección simple o compuesta dentro de estructuras de selección simple o compuesta. Esta anidación se puede hacer tantas veces como sea necesario.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G

Ejemplo 5: Estructuras de selección anidadas

ALGORITMO

```
algoritmo MenorMayorDeEdad2
variables
    entero: edad
inicio
    muestre('Entre su edad: ')
    lea(edad)
    si edad ≥ 0 entonces
        si edad ≥ 18 entonces
            muestre('Usted es mayor de edad')
        si_no
            muestre('Usted es menor de edad')
        fin_si
    si_no
        muestre('Error: No hay edades negativas')
    fin_si
fin
```

PROGRAMA

```
edad = input('Entre su edad: ');

if edad >= 0
    if edad >= 18
        disp('Usted es mayor de edad')
    else
        disp('Usted es menor de edad')
    end
else
    disp('Error: No hay edades negativas')
end
```

Ejemplo 6: Algoritmo y programa para calcular el área de un triángulo

ALGORITMO

```
algoritmo AreaTrianguloMejorado
variables
    real : base, altura, area
inicio
    muestre('CALCULAR EL AREA DE UN TRIANGULO')
    muestre('Entre la base: ')
    lea(base)
    si base > 0 entonces
        muestre('Entre la altura: ')
        lea(altura)
        si altura > 0 entonces
            area ← base * altura / 2
            muestre('El área es: ', area)
        si_no
            muestre('Error: Altura no valida')
        fin_si
    si_no
        muestre('Error: Base no valida')
    fin_si
fin
```

PROGRAMA

```
disp('CALCULAR EL AREA DE UN TRIANGULO')
base = input('Entre la base: ');
if base > 0
    altura = input('Entre la altura: ');
    if altura > 0
        area = base * altura / 2;
        disp('El area es: ')
        disp(area)
    else
        disp('Error: Altura no valida')
    end
else
    disp('Error: Base no valida')
end
```


Ejemplo 7: Algoritmo y programa para determinar el mayor de tres números

ALGORITMO

```
algoritmo MayorDeTresNumeros
variables
    real: a, b, c
inicio
    lea(a, b, c)
    si a > b entonces
        si a > c entonces
            muestre('El mayor es: ', a)
        si_no
            muestre('El mayor es: ', c)
        fin_si
    si_no
        si b > c entonces
            muestre('El mayor es: ', b)
        si_no
            muestre('El mayor es: ', c)
        fin_si
    fin_si
fin
```

PROGRAMA

```
a = input('Ingrese el primer numero: ');
b = input('Ingrese el segundo numero: ');
c = input('Ingrese el tercer numero: ');
if a > b
    if a > c
        disp('El mayor es: ')
        disp(a)
    else
        disp('El mayor es: ')
        disp(c)
    end
else
    if b > c
        disp('El mayor es: ')
        disp(b)
    else
        disp('El mayor es: ')
        disp(c)
    end
end
```


Ejemplo 8: Otra forma de solucionar el problema del ejemplo anterior

ALGORITMO

```
algoritmo MayorDeTresNumeros2
variables
    real: a, b, c
inicio
    lea(a, b, c)
    si  $a \geq b$  y  $a \geq c$  entonces muestre('El
        mayor es: ', a)
    si_no
        si  $b \geq a$  y  $b > c$  entonces muestre('El
            mayor es: ', b)
        si_no
            muestre('El mayor es: ', c) fin_si
    fin_si fin
```

PROGRAMA

```
a = input('Ingrese el primer numero: ');
b = input('Ingrese el segundo numero: ');
c = input('Ingrese el tercer numero: ');
if a >= b && a >= c
    disp('El mayor es: ')
    disp(a)
else
    if b >= a && b > c
        disp('El mayor es: ')
        disp(b)
    else
        disp('El mayor es: ')
        disp(c)
    end
end
```

Ejemplo 9: Algoritmo y programa para determinar el mayor de tres números

ALGORITMO

```
algoritmo MayorDeTresNumeros3
variables
    real: a, b, c, mayor
inicio
    lea(a, b, c)
    mayor ← a
    si b > mayor entonces
        mayor ← b
    fin_si
    si c > mayor entonces
        mayor ← c
    fin_si
    muestre('El mayor es: ', mayor) fin
```

PROGRAMA

```
a = input('Ingrese el primer numero: ');
b = input('Ingrese el segundo numero: ');
c = input('Ingrese el tercer numero: ');

mayor = a;
if b > mayor
    mayor = b;
end
if c > mayor
    mayor = c;
end

disp('El mayor es: ')
disp(mayor)
```

Ejemplo 10: Algoritmo y programa para determinar si un año es bisiesto

ALGORITMO

```
algoritmo AnioBisiesto
variables
    entero: anio
inicio
    muestre('Ingresa un año: ')
    lea(anio)
    si anio mod 4 = 0 entonces
        si anio mod 100 = 0 entonces si
            anio mod 400 = 0 entonces
                muestre('El año es bisiesto')
            si_no
                muestre('El año no es bisiesto')
        fin_si
    si_no
        muestre('El año es bisiesto')
    fin_si
    si_no
        muestre('El año no es bisiesto')
    fin_si
fin
```

PROGRAMA

```
anio = input('Ingresa un año: ');
if mod(anio, 4) == 0
    if mod(anio, 100) == 0
        if mod(anio, 400) == 0
            disp('El año es bisiesto ')
        else
            disp('El año no es bisiesto')
        end
    else
        disp('El año es bisiesto')
    end
else
    disp('El año no es bisiesto ')
end
```

Ejemplo 11: Otra forma de solucionar el problema del año bisiesto

ALGORITMO

```
algoritmo AnioBisiesto2
variables
    entero: anio
inicio
    muestre('Ingrese un año: ')
    lea(anio)
    si anio mod 4 = 0 y (anio mod 100
        ≠ 0 o anio mod 400 = 0) entonces
        muestre('El año es bisiesto')
    si_no
        muestre('El año no es bisiesto')
    fin_si
fin
```

PROGRAMA

```
anio = input('Ingrese un año: '); if
mod(anio, 4) == 0 &&
    (mod(anio, 100) ~= 0 ||
    mod(anio, 400) == 0)
    disp('El año es bisiesto ') else
    disp('El año no es bisiesto ') end
```



Ejemplo 12

Una empresa trabaja únicamente dos turnos, diurno y nocturno. El valor de la hora diurna es \$25000 y la nocturna \$32000, en caso que el trabajador labore el día domingo la tarifa diurna se incrementa un 2% del valor de la hora diurna y la tarifa nocturna se incrementa un 10% del valor de la hora nocturna. Por cada trabajador se conoce el día que laboró, la cantidad de horas laboradas y el turno. Diseñe un algoritmo que calcule el total a pagar por día al trabajador.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Algoritmo del ejemplo 12 (1/2)

algoritmo PagoTrabajadorPorDia

variables

real: horas, pago

entero: dia, turno

inicio

muestre('CALCULAR PAGO TRABAJADOR POR DIA')

muestre('Ingrese el día (1=Lunes, 2=Martes, ..., 7=Domingo): ')

lea(dia)

muestre('Ingrese las horas laboradas: ')

lea(horas)

muestre('Ingrese el turno (1=Diurno, 2=Nocturno): ')

lea(turno)

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Algoritmo del ejemplo 12 (2/2)

```
si turno = 1 entonces
    si dia = 7 entonces
        pago ← (25000 + 0.02 * 25000) * horas
    si_no
        pago ← 25000 * horas
    fin_si
si_no
    si dia = 7 entonces
        pago ← (32000 + 0.1 * 32000) * horas
    si_no
        pago ← 32000 * horas
    fin_si
fin_si
muestre('El total a pagar es: ', pago)
fin
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Estructura de Selección Múltiple

- Evaluará una expresión que podrá tomar n valores distintos. De acuerdo a la elección de uno de estos valores, se realizará una de la n acciones, es decir, el flujo del algoritmo seguirá un determinado camino entre los n posibles.
- Equivale a tener varias estructuras de selección compuesta anidadas.
- También es posible anidar estructuras de selección múltiple y combinarlas con estructuras de selección simple y compuesta.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Estructura de selección múltiple en pseudocódigo

```
⋮  
segun <variable> hacer  
    caso <Lista valores>  
        // Instrucciones  
    caso <Lista valores>  
        // Instrucciones  
    ⋮  
    en_otro_caso  
        // Instrucciones  
fin_segun  
⋮
```

- <variable>: Es la expresión a ser evaluada. La variable debe ser de tipo **entero** o **cadena**.
- <Lista valores>: Conjunto de valores posibles, separados por comas, para la <variable>.
- **caso**: Indica las posibles alternativas excluyentes de flujo del algoritmo de acuerdo a la lista de valores.
- **en_otro_caso**: En el evento que el valor de la variable no corresponda a alguno de los casos previos se realizarán, por defecto, las instrucciones establecidas aquí.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Estructura de selección múltiple en programas

```
switch <variable>
  case <Lista valores>
    % Instrucciones
  case <Lista valores>
    % Instrucciones
    ⋮
  otherwise
    % Instrucciones
end
```

<Lista valores>: El conjunto de valores debe ir separados por coma y entre llaves.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Ejemplo 13 : Calculadora básica


Diseñe un algoritmo que permite seleccionar una operación aritmética básica, leer dos números y mostrar el resultado de efectuar la operación sobre los dos números.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Ejemplo 13: Estructura de selección múltiple con variable de tipo **entero** (1/2)

algoritmo CalculadoraBasica **variables**

entero: operacion **real**: a, b,
resultado

inicio

muestre('CALCULADORA BASICA')

muestre('1. Suma') muestre('2. Resta')
muestre('3. Multiplicación') muestre('4.
División')

muestre('Elija la operación ingresando el numero de la opción: ') lea(operacion)

si operacion > 0 y operacion < 5 **entonces**

muestre('Ingrese el primer numero: ') lea(a)

muestre('Ingrese el segundo numero: ') lea(b)

fin_si

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Ejemplo 13: Estructura de selección múltiple con variable de tipo entero (2/2)

```
segun operacion hacer
  caso 1
    resultado ← a + b
    muestre('La suma es: ', resultado)
  caso 2
    resultado ← a - b
    muestre('La resta es: ', resultado)
  caso 3
    resultado ← a * b
    muestre('El producto es: ', resultado)
  caso 4
    si b ≠ 0 entonces
      resultado ← a / b
      muestre('El cociente es: ', resultado)
    si_no
      muestre('Error: La división por cero no esta definida.')
    fin_si
  en_otro_caso
    muestre('Error: Opción no valida') fin_segun
fin
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Ejemplo 14: Estructura de selección múltiple con variable de tipo **cadena** (1/2)

algoritmo AreaFiguraGeometrica

variables

real: area, radio, base, altura

cadena: figura

inicio

muestre('CALCULAR AREA FIGURA GEOMETRICA')

muestre('- Circulo')

muestre('- Rectagulo')

muestre('- Paralelogramo')

muestre('Elija una opción ingresando el nombre de la figura en minúsculas: ')

lea(figura)

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

Ejemplo 14: Estructura de selección múltiple con variable de tipo *cadena* (2/2)

```
segun figura hacer
  caso 'circulo'
    muestre('Ingrese el radio: ')
    lea(radio)
    area ← 3.1416 * radio ^ 2
    muestre('El área es: ', area)
  caso 'rectangulo', 'paralelogramo'
    muestre('Ingrese la base: ')
    lea(base)
    muestre('Ingrese la altura: ')
    lea(altura)
    area ← base * altura
    muestre('El área es: ', area)
  en_otro_caso
    muestre('Error: Opción no valida.')
fin_segun
fin
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Ejercicios

- Solucionar los ejemplos propuestos.
- Escriba el programa para el algoritmo del ejemplo cuatro y del ejemplo 12.
- Adicione otras operaciones al ejemplo de la calculadora básica.
- Adicione otras figuras geométricas al ejemplo donde se calculan sus áreas.
- Solucionar el taller 3.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Referencias

- Joyanes Aguilar, Luis. Fundamentos de Programación: Algoritmos, estructuras de datos y objetos, 4ª edición, Madrid: McGraw-Hill, 2008.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



■ WWW.MAKAIA.ORG

Info: comunicaciones@makaiia.org

Corporación MAKAIIA
Medellín, Colombia
Carrera 43A – 34-155.
Almacentro
Torre Norte, Oficina 701
Teléfono: (+574) 448 03 74
Móvil: (+57) 320 761 01 76



@makaiiaor