

PROGRAMACIÓN Y MÉTODOS NUMÉRICOS

2503506

ESTRUCTURAS DE DECISIÓN - Parte 1

Andrés Agudelo
Departamento de Ingeniería Mecánica
andres.agudelos@udea.edu.co



Introducción

- 1 Introducción
- 2 Estructura de selección simple
- 3 Estructura de selección compuesta
- 4 Tipos de condiciones
- 5 Anidamiento de estructuras de selección
- 6 A continuación

Contenido

- 1 Introducción
- 2 Estructura de selección simple
- 3 Estructura de selección compuesta
- 4 Tipos de condiciones
- 5 Anidamiento de estructuras de selección
- 6 A continuación

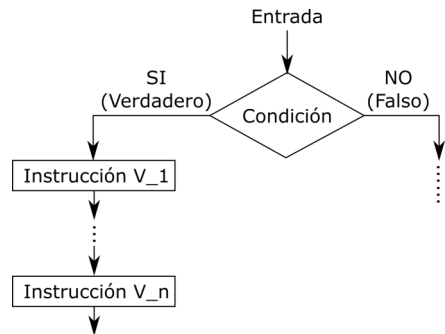
Introducción

Estructuras de selección

Se emplean cuando, en el desarrollo de la solución de un problema, se debe tomar una **decisión** que determina cual es el camino a seguir con base en una **condición**. → Estructuras de decisión.

- Tienen una entrada → Se compara con la condición.
- Su resultado (salida) puede ser sólo **verdadero** (Si = **True**) o **falso** (No = **False**).
- Crean una **bifurcación** del flujo del algoritmo. → Según el valor de la salida **se ejecuta sólo uno u otro** conjunto de instrucciones.

Estructuras de selección



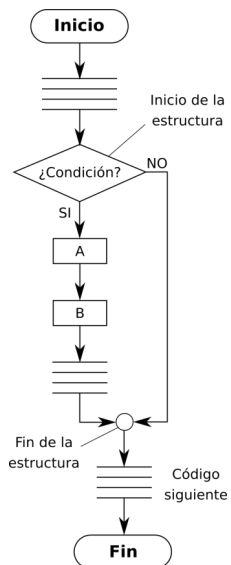
si condición **entonces**
 <Instrucción V_1>
 ...
 <Instrucción V_n>
 ...
fin si

Tipos de estructuras de selección

- 1 Simple → **si** ⇒ **entonces**
- 2 Compuesta → **si** ⇒ **entonces** / **sino**
- 3 Múltiple → **casos** (alternativas)

- 1 Introducción
- 2 Estructura de selección simple
- 3 Estructura de selección compuesta
- 4 Tipos de condiciones
- 5 Anidamiento de estructuras de selección
- 6 A continuación

Estructura de selección simple



algoritmo Estructura de selección simple

Entradas: Definición variables de entrada
Intermedias: Definición de variables auxiliares
Salidas: Definición de variables de salida

```

1 inicio
2 ...
3 si Condición entonces
4   Instrucción A
5   Instrucción B
6   ...
7   Instrucción n
8 fin si
9 Código siguiente
10 fin
  
```

```

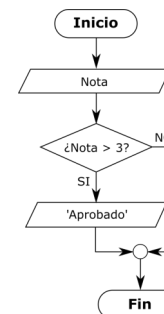
if condición:
    Instrucción A
    Instrucción B
    ...
    Instrucción n
Código siguiente
  
```

Estructura de selección simple

Ejemplo 1

Se pide leer una nota e imprimir un mensaje de aprobado si la nota es mayor que 3.

Diagrama de flujo



Pseudocódigo

algoritmo Nota

Entradas: real: Nota

```

1 inicio
2 leer Nota
3 si Nota > 3 entonces
4   escribir 'Aprobado'
5 fin si
6 fin
  
```

Python

```

Nota = float(input('Ingrese la nota: '))
if Nota > 3:
    print('Aprobado')
  
```

Estructura de selección simple

Ejemplo 1

Realizar la prueba de escritorio con las siguientes notas: 3.5, 2.9, −1.6, y 13.2.

Nota	Pantalla	Observación
3.5	'Aprobado'	Cuando la condición se cumple imprime el mensaje 'Aprobado'
2.9		Cuando la condición no se cumple va a buscar el fin de la estructura, y luego del algoritmo
−1.6		Un valor incorrecto de la nota no genera error
13.2	'Aprobado'	El algoritmo no establece un límite para la nota

1 Introducción

2 Estructura de selección simple

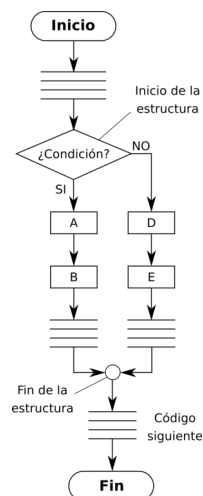
3 Estructura de selección compuesta

4 Tipos de condiciones

5 Anidamiento de estructuras de selección

6 A continuación

Estructura de selección compuesta



algoritmo Estr. de selección compuesta

Entradas: Definición variables de entrada

Intermedias: Definición de variables auxiliares

Salidas: Definición de variables de salida

```

1 inicio
2 ...
3 si Condición entonces
4     Instrucción A
5     Instrucción B
6     ...
7 sino
8     Instrucción D
9     Instrucción E
10    ...
11 fin si
12 Código siguiente
13 fin
  
```

```

if condición:
    Instrucción A
    Instrucción B
    ...
else:
    Instrucción D
    Instrucción E
    ...
Código siguiente
  
```

Ejemplo 2

A partir de las 3 notas de un curso, determinar si un alumno aprueba o reprueba, e imprimir el resultado. Todas las notas tienen igual valor (1/3), y la nota mínima aprobatoria es 3.

Análisis: Qué y cómo → Variables y operaciones

Nota final = promedio(Notas): Prom

Aprueba si la nota final es mayor o igual a 3.

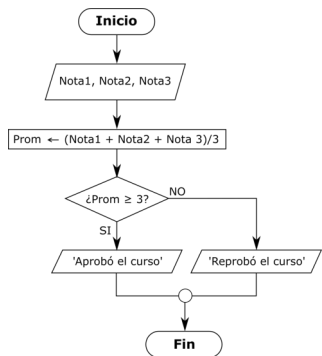
Notas: Nota1, Nota2, Nota3

$Prom \leftarrow (Nota1 + Nota2 + Nota3)/3$

Estructura de selección compuesta

Estructura de selección compuesta

Diagrama de flujo



Pseudocódigo

algoritmo Nota final de un

curso

Entradas: real: Nota1, Nota2, Nota3

var final real: Prom ; %Nota

```

1 inicio
2   leer Nota1, Nota2, Nota3
3   Prom <- (Nota1 + Nota2 +
4     Nota3)/3
5   si Prom >= 3 entonces
6     escribir 'Aprobó el curso'
7   sino
8     escribir 'Reprobó el
9     curso'
10  fin si
11 fin
  
```

Python

algoritmo Nota final de un curso

```

# Entradas:
Nota1 = float(input('Nota 1: '))
Nota2 = float(input('Nota 2: '))
Nota3 = float(input('Nota 3: '))

# Cálculos:
Prom = (Nota1 + Nota2 + Nota3) / 3

# Salidas:
if Prom >= 3:
    print('Aprobó el curso')
else:
    print('Reprobó el curso')
  
```

Estructura de selección compuesta

Ejemplo 2

Realizar la prueba de escritorio con las siguientes notas: (3.1, 4.5, 2.7), (2.7, 3.2, 2.3), (−7.4, 2.1, 3.5), y (3.7, 9.2, 4.3).

Nota1	Nota2	Nota3	Prom	Pantalla
3.1	4.5	2.7		
3.1	4.5	2.7	3.43	
3.1	4.5	2.7	3.43	'Aprobó el curso'
2.7	3.2	2.3		
2.7	3.2	2.3	2.73	
2.7	3.2	2.3	2.73	'Reprobó el curso'
−7.4	2.1	3.5		
−7.4	2.1	3.5	−0.6	
−7.4	2.1	3.5	−0.6	'Reprobó el curso'
3.7	9.2	4.3		
3.7	9.2	4.3	5.73	
3.7	9.2	4.3	5.73	'Aprobó el curso'

Ejemplo 3 – Cálculo de salario con horas extraordinarias

algoritmo Cálculo del salario

Entradas: str: nombre

entero: horas, valor_ho

Intermedias: real: S_ord, S_ext, imp

Salidas: real: S_bruto, S_netto

string: nombre

```

1 inicio
2   leer nombre, horas, valor_ho
3   si horas > 40 entonces
4     S_ord <- 40 * valor_ho
5     S_ext <- 1.5 * valor_ho * (horas - 40)
6     S_bruto <- S_ord + S_ext
7     imp <- 0.25 * S_ord + 0.1 * S_ext
8   sino
9     S_ord <- horas * valor_ho
10    S_bruto <- S_ord
11    imp <- 0.25 * S_bruto
12  fin si
13  S_netto <- S_bruto - imp
14  escribir 'Nombre: ', nombre
15  escribir 'Salario bruto: $', S_bruto
16  escribir 'Salario netto: $', S_netto
17 fin
  
```

```

# Entrada de datos:
nombre = input('Ingrese el nombre del trabajador: ')
horas = int(input('Ingrese el número de horas trabajadas en la semana: '))
valor_ho = int(input('Ingrese el valor de la hora ordinaria: '))

# Cálculos:
if horas > 40:
    S_ord = 40 * valor_ho
    S_ext = 1.5 * valor_ho * (horas - 40)
    S_bruto = S_ord + S_ext
    imp = 0.25 * S_ord + 0.1 * S_ext
else:
    S_bruto = horas * valor_ho
    imp = 0.25 * S_bruto
S_netto = S_bruto - imp

# Salida (resultados):
print('\nNombre: ', nombre)
print('Salario bruto: $', S_bruto, sep='')
print('Salario netto: $', S_netto, sep='')
  
```

1 Introducción

2 Estructura de selección simple

3 Estructura de selección compuesta

4 Tipos de condiciones

5 Anidamiento de estructuras de selección

6 A continuación

Tipos de condiciones

Tipos de condiciones

Las condiciones pueden ser únicas o se puede usar una serie de condiciones. En cualquier caso, el resultado final debe tener un valor lógico que puede ser sólo **verdadero** o **falso**.

- **Condición única:**
si condición **entonces**
- **Serie de condiciones:** Se usan operadores de relación
 - Comparaciones sencillas
si (cond 1 **y** cond 2 **y** ...) **entonces**
si (cond 1 **o** cond 2 **o** ...) **entonces**
 - Comparaciones compuestas
si (cond 1 **y** cond 2) **o** (cond 3 **y** cond 4) **entonces**

Estructuras de selección simple y compuesta

algoritmo Estructuras de selección simple y compuesta

Entradas: Definición variables de entrada

Intermedias: Definición de variables auxiliares

Salidas: Definición de variables de salida

```

1  inicio
2  ...
3  si (Condición 1 y Condición 2 ...) entonces
4  | Instrucciones
5  | ...
6  sino
7  | Instrucciones
8  | ...
9  fin si
10 ...
11 fin
  
```

Anidamiento de estructuras de selección

- 1 Introducción
- 2 Estructura de selección simple
- 3 Estructura de selección compuesta
- 4 Tipos de condiciones
- 5 Anidamiento de estructuras de selección
- 6 A continuación

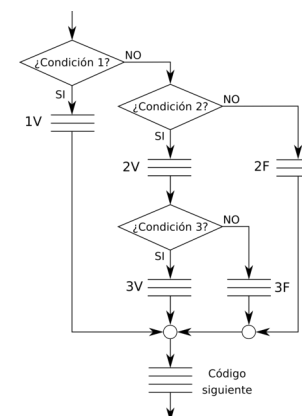
Anidamiento de estructuras de selección

Anidamiento de estructuras de selección

Anidamiento de estructuras de selección

Se presenta cuando el resultado de una estructura de selección (simple o compuesta) lleva a la evaluación de otra condición.

Esta situación se puede repetir varias veces.



```

si condición 1 entonces
  Instrucciones 1V
sino
  si condición 2 entonces
    Instrucciones 2V
    si condición 3 entonces
      Instrucciones 3V
    sino
      Instrucciones 3F
  fin si
  sino
    Instrucciones 2F
  fin si
fin si
Código siguiente
  
```

```

if condición 1:
  Instrucciones 1V
elif condición 2:
  Instrucciones 2V
  if condición 3:
    Instrucciones 3V
  else:
    Instrucciones 3F
else:
  Instrucciones 2F
Código siguiente
  
```

Anidamiento de estructuras de selección

Ejemplo 4

Durante la celebración de aniversario de un almacén se conceden descuentos a los clientes, de acuerdo con el color de una balota que éstos sacan antes de pagar la cuenta total.

Se tiene un saco con balotas de tres colores:

- Blanco → No se hace descuento.
- Verde → Se hace un descuento del 10 %.
- Rojo → Se hace un descuento del 20 %.

Escriba un algoritmo para determinar el valor final de la cuenta, después de incluir el descuento obtenido con las balotas.

Sólo existen tres colores válidos para los descuentos, si aparece uno diferente, o si el cajero se equivoca al introducirlo, se debe detectar este error.

Anidamiento de estructuras de selección

Ejemplo 4 - Análisis

A los clientes que van a pagar se les permite sacar una balota de un color determinado, y se aplica un descuento a su compra en función del color de ésta.



Hace falta saber cuanto debía pagar el cliente originalmente, así como el color de la balota que sacó.

En función del color se debe calcular el valor del descuento, según el costo de la compra.

El objetivo del algoritmo es determinar el pago final que debe realizar el cliente, a partir del pago original, y de la balota que sacó.

Anidamiento de estructuras de selección

Ejemplo 4

Planteamiento del problema:

- Variables de entrada:
 - Costo total de la compra → `ctotal`
 - Color de la balota que saca el cliente → `color`.
- Variables de salida:
 - Pago final que debe realizar el cliente, después de aplicar el descuento ganado en la actividad → `pago`.
- Variables intermedias:
 - Valor del descuento correspondiente al color de la balota → `desc`.
 - Identificador de error en caso de que se presente una balota que tiene un color no establecido → `error`.

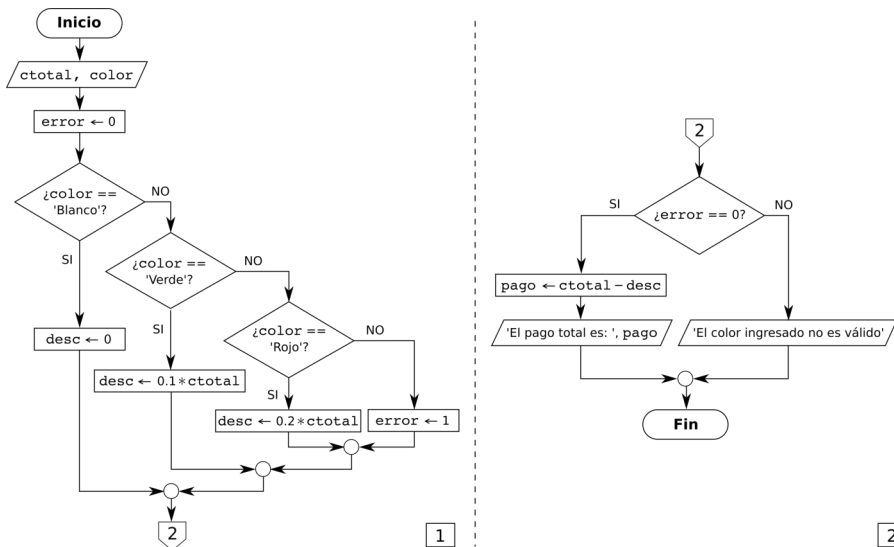
$\text{color} \Rightarrow \text{desc} = \text{ctotal} * x \quad (x = 0, 0.1, 0.2)$
 $\text{pago} = \text{ctotal} - \text{desc}$
 $\text{error: } 0 \Rightarrow \text{No hay error, } 1 \Rightarrow \text{Hay error.}$

Anidamiento de estructuras de selección

Ejemplo 4 - Pasos

- 1 Leer los valores de `ctotal` y de `color`.
- 2 Inicializar `error` en cero.
- 3 Identificar el color de la balota:
 - Blanco: No hay descuento $\Rightarrow \text{desc} \leftarrow 0$
 - Verde: Descuento del 10 % $\Rightarrow \text{desc} \leftarrow 0.1 * \text{ctotal}$
 - Rojo: Descuento del 20 % $\Rightarrow \text{desc} \leftarrow 0.2 * \text{ctotal}$
 - Color no válido: Existe error $\Rightarrow \text{error} \leftarrow 1$
- 4 Calcular el pago final con descuento:
 - Si no hay error en el color de la balota:
 $(\text{error} == 0) \Rightarrow \text{pago} \leftarrow \text{ctotal} - \text{desc}$
 Mostrar el resultado mediante un mensaje ('El pago total es: ', `pago`)
 - Error de color: Escribir un mensaje que identifique el error ('El color ingresado no es válido')

Anidamiento de estructuras de selección



algoritmo Cálculo de descuento

Entradas: real: ctotal string: color
 Intermedias: real: desc entero: error
 Salidas: real: pago

```

1 inicio
2   error ← 0
3   leer ctotal, color
4   si color == 'Blanco' entonces
5     desc ← 0
6   sino
7     si color == 'Verde' entonces
8       desc ← ctotal * 0.1
9     sino
10      si color == 'Rojo' entonces
11        desc ← ctotal * 0.2
12      sino
13        error ← 1
14      fin si
15    fin si
16  fin si
17  si error == 0 entonces
18    pago ← ctotal - desc
19    escribir 'El pago total es: ', pago
20  sino
21    escribir 'El color ingresado no es válido'
22  fin si
23 fin
  
```

```

# Inicialización del error:
error = 0

# Entrada de datos:
ctotal = float(input('Ingrese el costo total: '))
color = input('Ingrese el color de la balota: ')

# Identificación del color de la balota:
if color == 'Blanco':
    desc = 0
elif color == 'Verde':
    desc = ctotal * 0.1
elif color == 'Rojo':
    desc = ctotal * 0.2
else:
    error = 1

# Verificación del color y cálculo del pago
if error == 0:
    pago = ctotal - desc
    print('El pago total es:', pago)
else:
    print('El color ingresado no es válido')
  
```

Anidamiento de estructuras de selección

Ejemplo 4

Realizar la prueba de escritorio con los siguientes valores de CTotal y de Color:

(380,000; Verde), (50,000; Blanco), (123,000; Rojo), (400,000; Amarillo)

CTotal	Color	Error	Desc	Pago	Pantalla
		0			
380,000	'Verde'	0			
380,000	'Verde'	0	38,000		
380,000	'Verde'	0	38,000	342,000	
380,000	'Verde'	0	38,000	342,000	'El pago total es: 342,000'
50,000	'Blanco'	0			
50,000	'Blanco'	0	0		
50,000	'Blanco'	0	0	50,000	
50,000	'Blanco'	0	0	50,000	'El pago total es: 50,000'

Anidamiento de estructuras de selección

Ejemplo 4

Realizar la prueba de escritorio con los siguientes valores de CTotal y de Color:

(380,000; Verde), (50,000; Blanco), (123,000; Rojo), (400,000; Amarillo)

CTotal	Color	Error	Desc	Pago	Pantalla
123,000	'Rojo'	0			
123,000	'Rojo'	0	24,600		
123,000	'Rojo'	0	24,600	98,400	
123,000	'Rojo'	0	24,600	98,400	'El pago total es: 98,400'
		0			
400,000	'Amarillo'	0			
400,000	'Amarillo'	1			
400,000	'Amarillo'	1			'El color ingresado no es válido'

Anidamiento de estructuras de selección

Ejemplo 5

El problema de calcular el descuento en el ejemplo 3 se resolvió usando el tipo de condición única.

- 1 Identifique posibles causas de que el algoritmo funcione mal.
- 2 Realice un algoritmo que resuelva el mismo problema usando una serie de condiciones.
- 3 Señale los beneficios de implementar la solución de esta manera.

Análisis:

El error en el color de la balota se da cuando ésta no es blanca, ni verde, ni roja \Rightarrow Después de leer el color, se puede verificar si éste es alguno de esos tres colores, usando una serie de condiciones:

$\text{if } \text{color} == ('Blanco' \text{ o } 'Verde' \text{ o } 'Rojo')$?

algoritmo Cálculo de descuento

```

Entradas:    real: cttotal    string: color
Intermedias: real: desc
Salidas:    real: pago

1 inicio
2   error ← 0
3   leer cttotal, color
4   si color == 'Blanco' entonces
5       | desc ← 0
6   sino
7       | si color == 'Verde' entonces
8       | | desc ← cttotal * 0.1
9       | sino
10          | si color == 'Rojo' entonces
11          | | desc ← cttotal * 0.2
12          | sino
13          | | error ← 1
14          | fin si
15          fin si
16   fin si
17   si error == 0 entonces
18       | pago ← cttotal - desc
19       | escribir 'El pago total es: ', pago
20   sino
21       | escribir 'El color ingresado no es válido'
22   fin si
23 fin
  
```

algoritmo Cálculo de descuento

```

Entradas:    real: cttotal    string: color
Intermedias: real: desc
Salidas:    real: pago

1 inicio
2   leer cttotal, color
3   si color == ('Blanco' o 'Verde' o 'Rojo') entonces
4       | si color == 'Blanco' entonces
5       | | desc ← 0
6       | sino
7       | | si color == 'Verde' entonces
8       | | | desc ← cttotal * 0.1
9       | | sino
10          | | | desc ← cttotal * 0.2
11          | | fin si
12       | fin si
13       | pago ← cttotal - desc
14       | escribir 'El pago total es: ', pago
15   sino
16       | escribir 'El color ingresado no es válido'
17   fin si
18 fin
  
```

algoritmo Cálculo de descuento

```

Entradas:    real: cttotal    string: color
Intermedias: real: desc
Salidas:    real: pago

1 inicio
2   leer cttotal, color
3   si color == ('Blanco' o 'Verde' o 'Rojo') entonces
4       | si color == 'Blanco' entonces
5       | | desc ← 0
6       | sino
7       | | si color == 'Verde' entonces
8       | | | desc ← cttotal * 0.1
9       | | sino
10          | | | desc ← cttotal * 0.2
11          | | fin si
12       | fin si
13       | pago ← cttotal - desc
14       | escribir 'El pago total es: ', pago
15   sino
16       | escribir 'El color ingresado no es válido'
17   fin si
18 fin
  
```

```

# Entrada de datos:
cttotal = float(input('Ingrese el costo total: '))
color = input('Ingrese el color de la balota: ')

# Identificación del color de la balota:
if (color == 'Blanco') or (color == 'Verde') or (color == 'Rojo'):
    if color == 'Blanco':
        desc = 0
    elif color == 'Verde':
        desc = cttotal * 0.1
    else:
        desc = cttotal * 0.2
    pago = pago + cttotal - desc
    print('El pago total es: ', pago)
else:
    print('El color ingresado no es válido')
  
```

A continuación

Próxima clase

Estructuras de decisión:

- Estructura de decisión múltiple.
- Ejemplos.
- Ejercicios.