

Slide 1

TEMA 6: ESTRUCTURAS BÁSICAS DE CONTROL	
<b>Índice</b>	
5.1. Estructura secuencial . . . . .	3
5.2. Estructuras alternativas . . . . .	3
5.2.1. Alternativa simple . . . . .	4
5.2.2. Alternativa doble . . . . .	6
5.3. Estructura iterativa . . . . .	9
5.3.1. Con condición al final . . . . .	9
5.3.2. Elección de sentencia iterativa . . . . .	9
5.3.3. Validación de bucles . . . . .	9
5.3.4. Control de bucles . . . . .	9

Slide 2

PSEUDOCÓDIGO	
■ Acciones	
■ Estructuras de control	
● Secuencia	
● Selección	
○ simple ( <b>si</b> ... <b>entonces</b> ...)	
○ doble ( <b>si</b> ... <b>entonces</b> ... <b>si no</b> ...)	
○ múltiple ( <b>según el caso</b> ...)	
● Iteración	
○ con condición al principio ( <b>mientras</b> ... <b>hacer</b> ...)	
○ con condición al final ( <b>repetir</b> ... <b>hasta que</b> ...)	
○ controlada por contador ( <b>para</b> ...)	
○ ...	

Slide 3

```

<sentencia> ::= <sentencia simple> | <sentencia compuesta>
<sentencia simple> ::= <sentencia de asignación> |
    read ... |
    write ... |
    <sentencia vacía>
<sentencia compuesta> ::= <sentencia secuencial> |
    <sentencia alternativa> |
    <sentencia iterativa>

```

Slide 4

```

                SECUENCIA
                <acción>
                <acción>
                ...
                <acción>


---


<sentencia secuencial> ::=
    begin
        <sentencia>
        {;<sentencia> }
    end

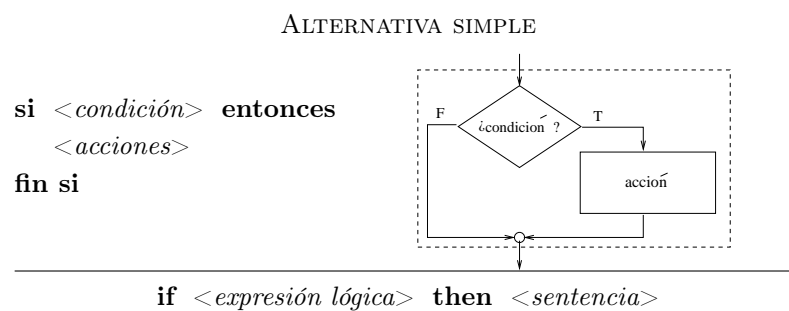
```

Slide 5

- Selección
  - simple (**si ... entonces ...**)
  - doble (**si ... entonces ... si no ...**)
  - múltiple (**según el caso ...**)

$\langle \text{sentencia alternativa} \rangle ::= \langle \text{alternativa simple} \rangle \mid$   
 $\langle \text{alternativa doble} \rangle \mid$   
 $\langle \text{alternativa múltiple} \rangle$

Slide 6



Slide 7

```

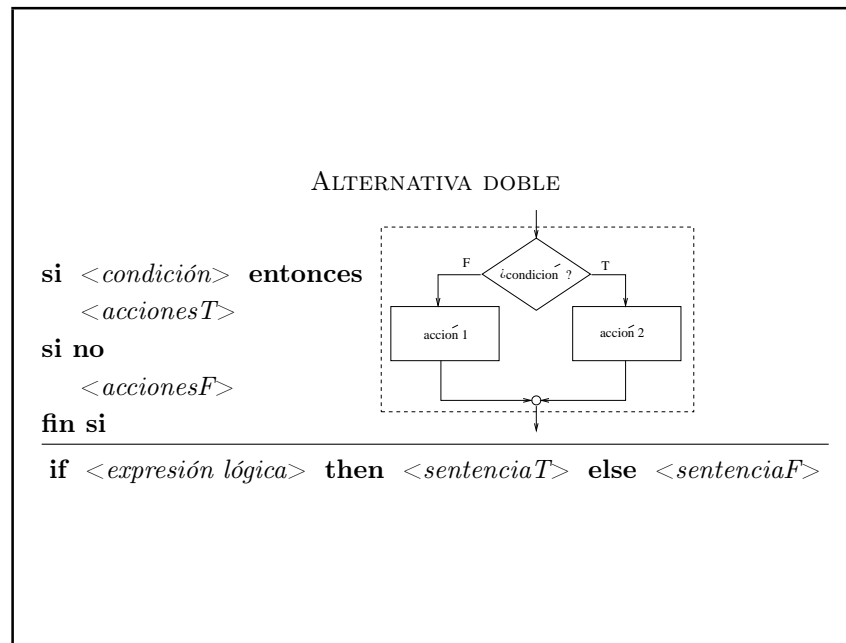
(* Pre: a, b, c > 0 *)
EsTriangulo := (a < b+c) and (b < a+c) and (c < a+b) ;
EsEquilatero := (a = b) and (b = c);
EsIsosceles := (a = b) and (a <> c) or
               (a = c) and (a <> b) or
               (b = c) and (a <> b);
EsEscaleno := not EsEquilatero and not EsIsosceles;
if EsTriangulo then writeln ('Forma un triángulo');
if EsTriangulo and EsEquilatero then writeln ('Equilát.');
```

Slide 8

```

(* Pre: a, b, c > 0 *)
EsTriangulo := (a < b+c) and (b < a+c) and (c < a+b) ;
if EsTriangulo then
  begin
    writeln ('Forman un triángulo');
    if (a=b) and (b=c) then writeln ('Equilátero');
    if (a=b) and (b<>c) or
       (a=c) and (b<>c) or
       (b=c) and (a<>b) then writeln ('Isósceles');
    if (a<>b) and (a<>c) and (b<>c) then writeln ('Esc.')
```

Slide 9



Slide 10

```

if a<0 then
  a := a+1
else
  a := a-1

```

```

if a<0 then
  a := a+1 ;
if a>=0 then
  a := a-1

```

```

if a>= 0 then
  a := a-1;
if a<0 then
  a:= a+1

```

Slide 11

```

if a <> 0 then (* ecuación de 2º grado *)
  if sqr(b)-4*a*c > 0 then
    writeln (output, '2 raíces reales')
  else (* discriminante <= 0 *)
    if sqr(b)-4*a*c = 0 then
      writeln (output, '1 raíz doble')
    else
      writeln (output, '2 raíces complejas conjugadas')
  else (* a=0, ecuación de grado < 2 *)
    if b <> 0 then
      writeln (output, '1 solución')
    else (* ecuación de grado 0, c=0 *)
      if c = 0 then
        writeln (output, 'cualquier número es solución')
      else
        writeln (output, 'no hay solución')

```

Slide 12

```

(* Pre: a, b, c > 0 *)
if (a<b+c) and (b<a+c) and (c<a+b) then
  begin
    write ('Forman triángulo ');
    if a = b then
      if a = c then writeln ('equilátero')
      else (* a=b<>c *)
        writeln ('isósceles')
    else (* a<>b *)
      if a = c then writeln ('isósceles')
      else (* a<>c, a<>b *)
        if b = c then writeln ('isósceles')
        else (* a<>c a<>b b<>c *)
          writeln ('escaleno')
    end
  else writeln ('No forman triángulo')

```

Slide 13

```
(* Pre: a, b, c > 0 *)
if (a<b+c) and (b<a+c) and (c<a+b) then
  begin
    write ('Forman triángulo ');
    if (a=b) and (b=c) then
      writeln ('equilátero')
    else if (a=b) or (b=c) or (a=c) then
      writeln ('isósceles')
    else
      writeln ('escaleno')
  end
else writeln ('No forman triángulo')
```