

Edith Clarke (1883-1959)



- Graduada en matemáticas e astronomía (1908)
- Calculadora (1912) en American Telephone and Telegraph (ATT)
- Creadora dunha calculadora gráfica patentada en 1925
- Enxeñeira eléctrica estadounidense e profesora de matemáticas, física e enxeñaría eléctrica

Sentenzas de selección

- Programa: execución secuencial de instruccións (*print*, *read*, asignacións, ...)
- Sentenza de selección: executa bifurcacións na execución secuencial do programa
- Avalía unha expresión lóxica: se se cumpre, executa unhas sentenzas; se non, executa outras ou non fai nada
- Permite obter programas complicados, con múltiples rutas de execución dependendo de datos de entrada
- Moi importantes na programación: permiten controla-lo fluxo de execución do programa

if lóxico / bloque *if*

- ***if* lóxico**: avalía unha condición (expresión lóxica) e dependendo do seu valor (*.true./false.*) executa ou non unha única sentenza

```
if(condición) sentenza
```

- A condición vai entre parénteses. Só válido cunha única sentenza. Se non se cumpre, non fai nada.

- **Bloque *if***: igual, pero permite varias sentenzas

```
if(condición) then  
    sentenzas  
end if
```

```
if(x >= 10.or.y /= -5) then  
    x = x+y; print *, x  
end if
```

- Atención ao *then*: se falta, erro de compilación

Bloque *if/else*

- Avalía a condición: se esta se cumpre, executa un bloque de sentenzas; se non se cumpre a condición, executa outro bloque distinto

```
if(condición) then  
    sentenzas1  
else  
    sentenzas2  
end if
```

```
if(x==0.or.y<-1) then  
    x=x+y;print *, x  
else  
    y = y - x; print *, y  
end if
```

- As sentenzas de selección poden conter outras sentenzas de selección, formando unha estrutura complexa.

Bloque *if/else* múltiple

- Avalía N condiciones: cada una tiene un bloque de sentencias, que se ejecuta si se cumple esa condición (entonces ya no se evalúa ninguna otra condición)
- A i -ésima condición solo se evalúa si no se cumple la condición $(i-1)$ -ésima (ni las i condiciones anteriores)
- Opcionalmente, puede haber un bloque $N+1$ que se ejecuta si ninguna condición se cumple

```
if(condición1) then  
    sentenzas1  
  
    else if(condición2) then  
        sentenzas2  
  
    ...  
  
    else if(condiciónN) then  
        sentenzasN  
  
    else  
        sentenzasN+1  
  
    endif
```

Sentenza *select*

- Compara secuencialmente unha *expresión enteira* con N selectores: valor único (0), conxunto de valores (1,2,3) ou rango de valores (:1)
- Se a expresión é igual a algún selector, execútase ese bloque e remátase
- So se executa un bloque
- Pode haber opcionalmente un bloque *default* que se executará se a expresión non se corresponde con ningún selector

```
select case (expresión)  
  case (selector1)  
    sentenzas1  
  case (selector2)  
    sentenzas2  
  
  ...  
  case (selectorN)  
    sentenzasN  
  case default  
    sentenzasN+1  
  
end select
```

Exemplos de *if/else* e *select*

- Exemplo de *if/else* múltiple: función a cachos:

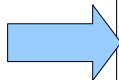
$$f(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$



```
if(x < 0) then
  f = -1
else if(x == 0) then
  f = 0
else
  f = 1
end if
```

- Exemplo de *select*:

código de
operación:
(*n* inteiro)



```
select case (n)
case (-1,0,1)
  print *, 'erro'
case (2:5)
  print *, 'correcto'
case default
  print *, 'apagado'
end select
```