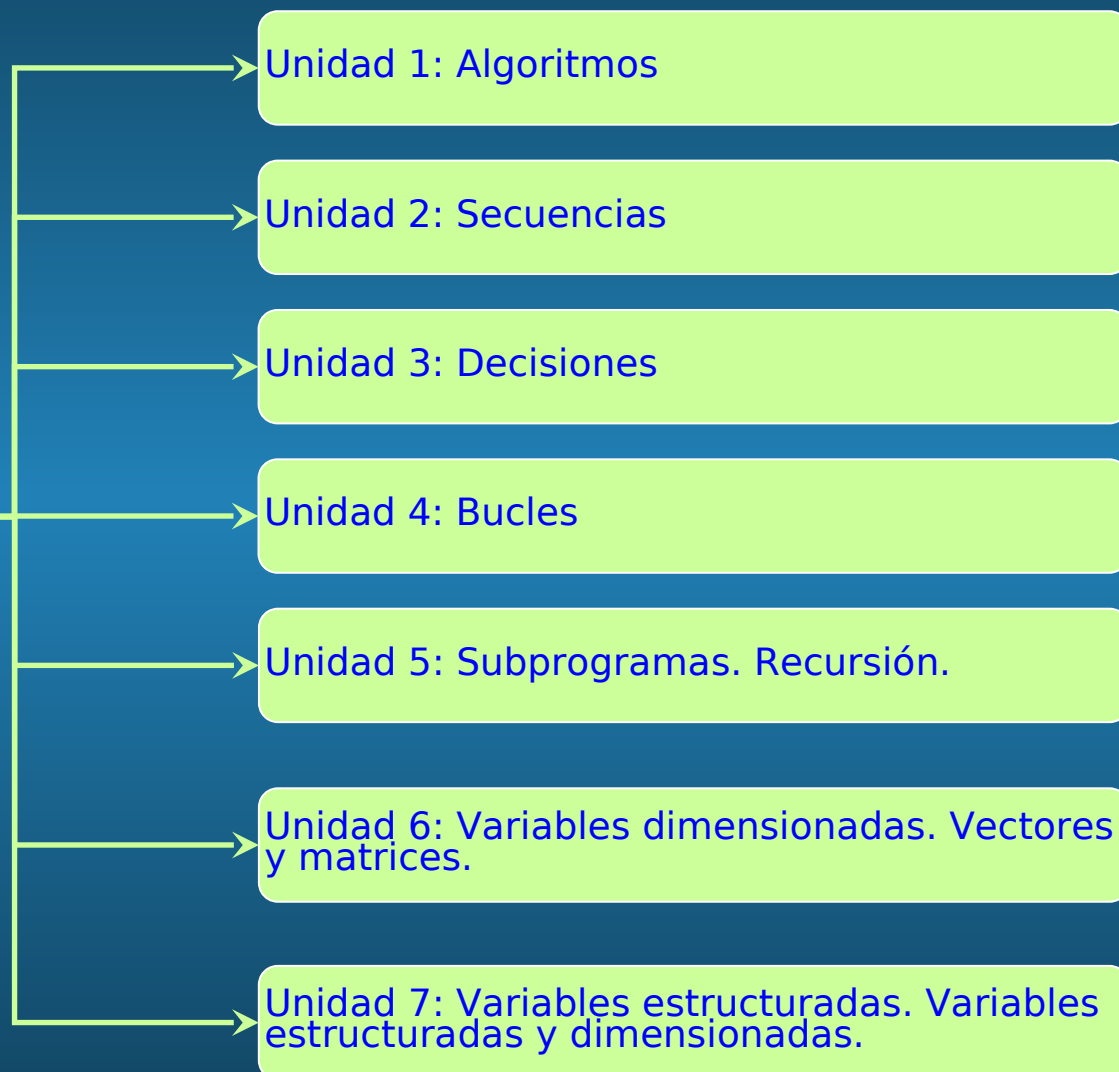




## Programación I

Con pseudocódigo





Por Composición de Alternativas Anidadas.

- **Conclusiones:**

- El orden de evaluación *modifica la eficiencia de la estructura.*
- ▮ Las condiciones escritas son *dependientes del orden de evaluación.*





Por Composición de Alternativas Anidadas.

- Para reordenarlas habría que *llevarlas a una forma mutuamente excluyente*.
- Para eso es necesario crear – *a partir del juego de condiciones originales* – un nuevo juego de **condiciones** (que llamaremos condiciones “*prima*”, como  $C_i'$ ) que sí sean mutuamente excluyentes:

$$\begin{aligned} C_1' &= C_1; \quad C_2' = \neg C_1 \wedge C_2; \quad C_i' = \neg C_1 \wedge \dots \wedge \neg C_{i-1} \wedge C_i; \\ C_n' &= \neg C_1 \wedge \dots \wedge C_n; \quad C_{n+1}' = \neg C_1 \wedge \dots \wedge C_n \end{aligned}$$



Por Composición de Alternativas Anidadas.

- Recién entonces puede procederse a la **reescritura completa** del **nuevo** juego de condiciones, *en cualquier orden*, aprovechando que **son excluyentes entre sí**.

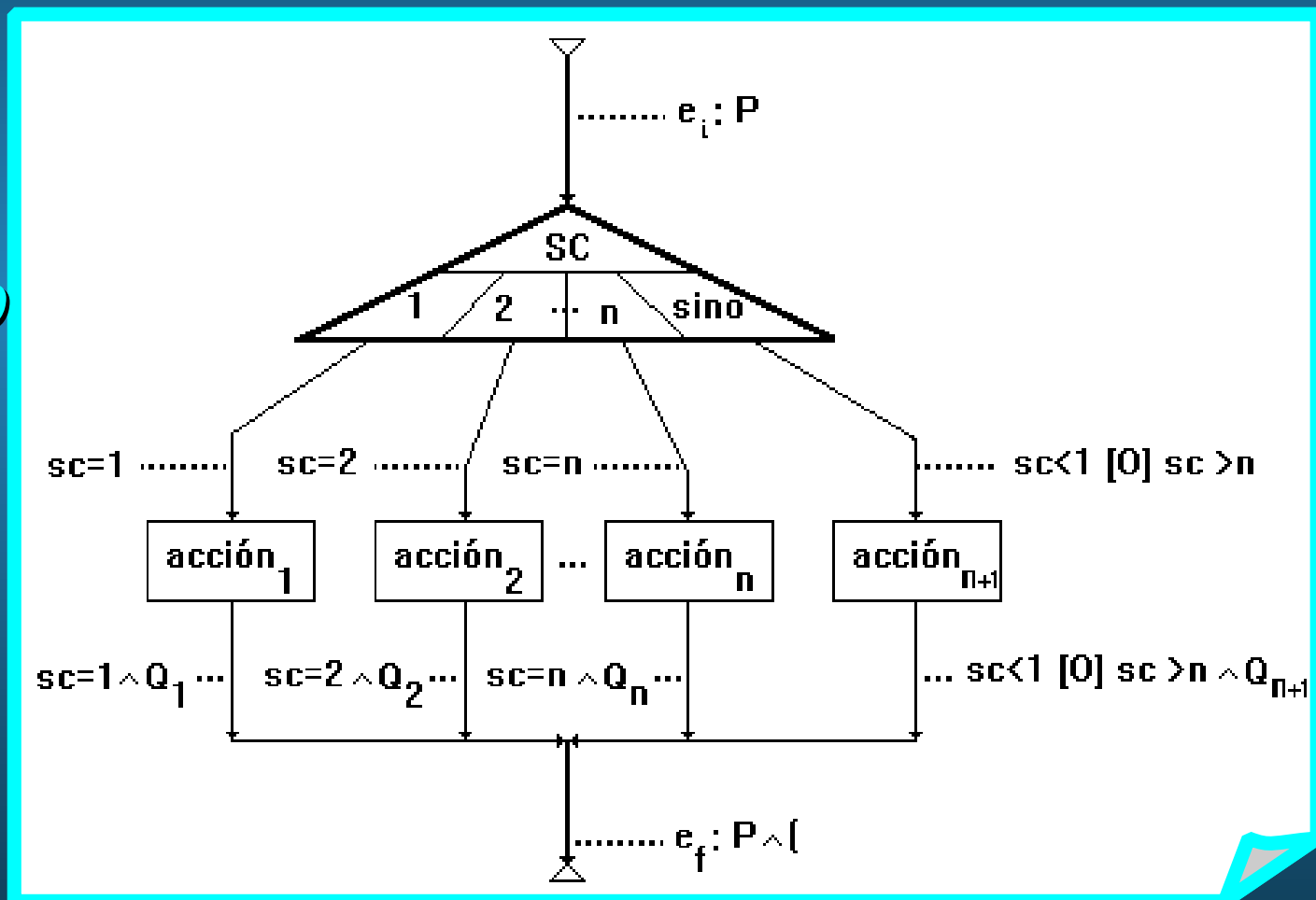
$$e_f: P \wedge ((C'_1 \wedge Q_1) \vee (C'_2 \wedge Q_2) \vee \dots (C'_n \wedge Q_n) \vee (C'_{n+1} \wedge Q_{n+1}))$$

- Desde la segunda condición en adelante se **debe aprovechar** lo que **ya se sabe** al haber sido **falsas las condiciones anteriores** para *evitar redundancia en las condiciones*.



Por Composición Selectiva.

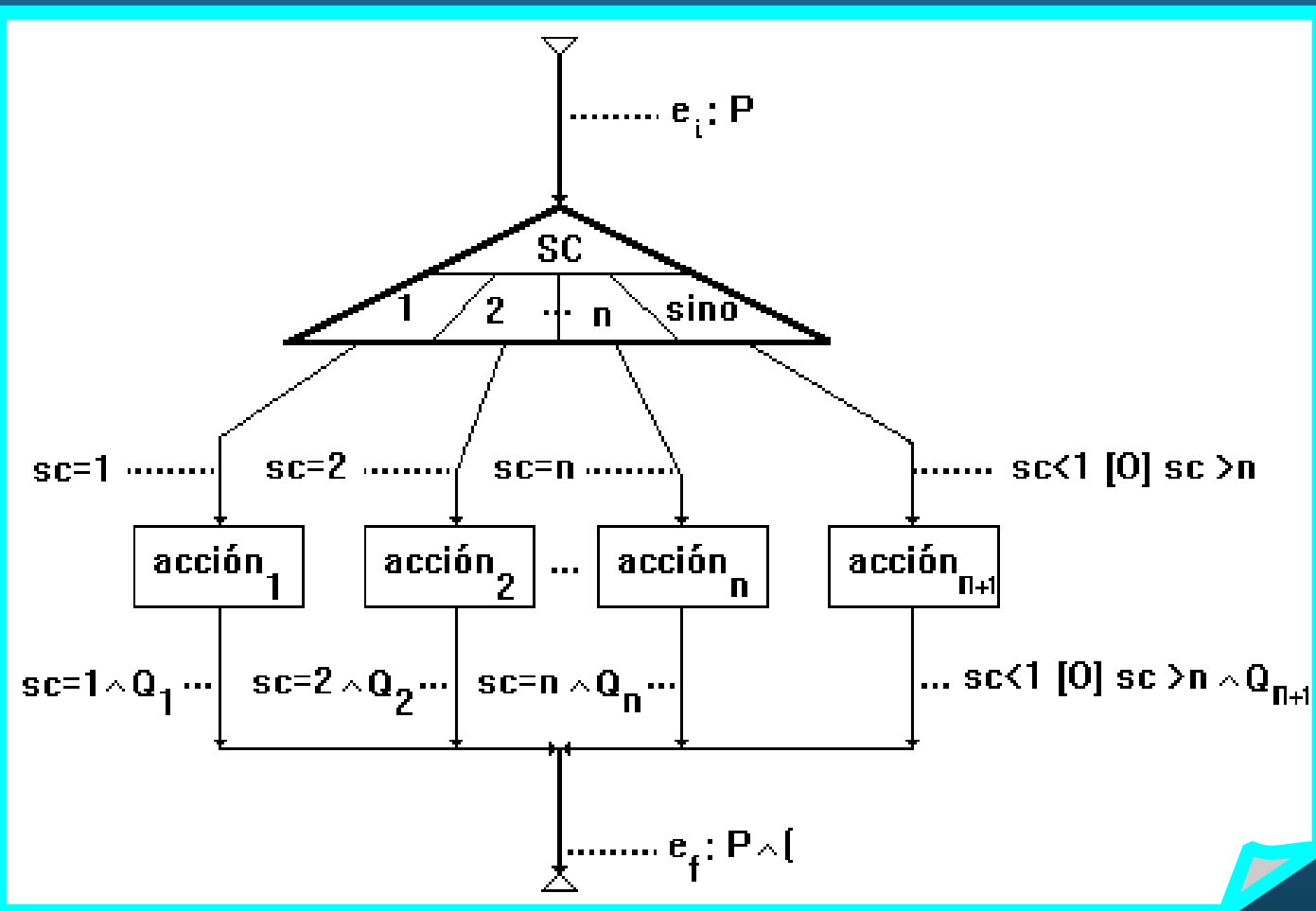
- Hay casos donde *naturalmente* las condiciones a evaluar son mutuamente excluyentes, como *cuando se pregunta por valores enteros sucesivos* (selector de caso).





## Por Composición Selectiva.

- Se utiliza una **expresión entera** como selectora.
- El orden de evaluación **no modifica la eficiencia de la estructura.**



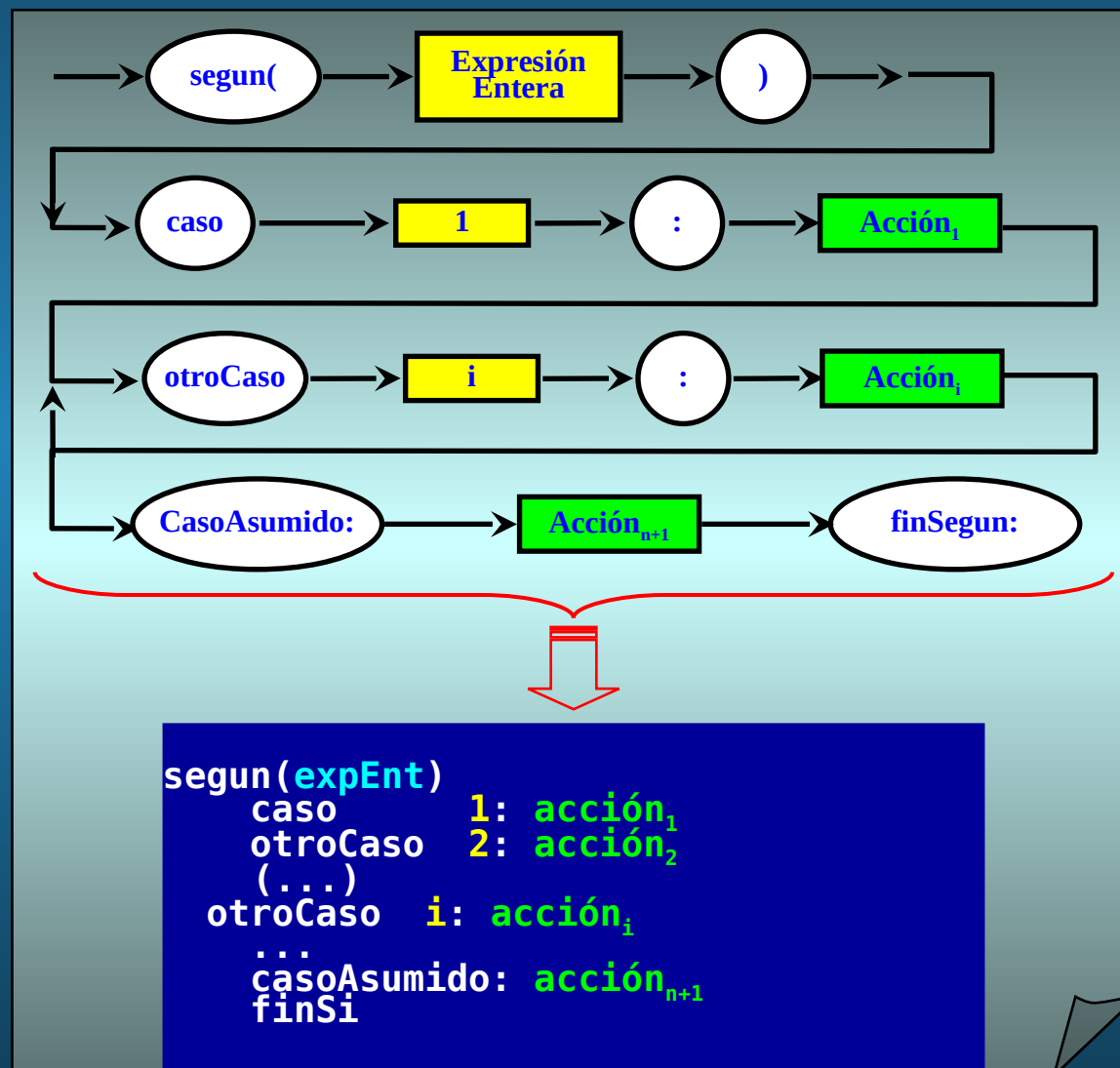


## Por Composición Selectiva.

### ○ Sintaxis:

▮ Se utiliza una **expresión numérica entera** como *llave selectora*.

▮ Después del primer caso debe usarse **otroCaso**.





## Enunciado:

- **Dados 3 valores, mostrarlos de menor a mayor.**

## Análisis:

- Los datos **deben** leerse y guardarse en 3 variables **numéricas** reales.
- Debe asegurarse el mostrar los 3 valores en orden creciente.
  - ▮ Empezaremos por el menor, mostrándolo **primero**.
  - ▮ Luego vendrán el del **medio** y el **mayor**.

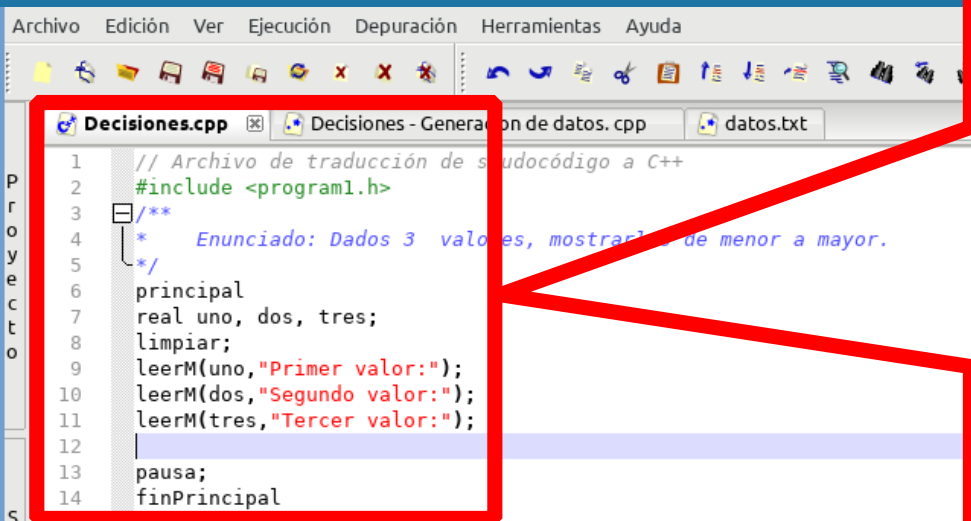




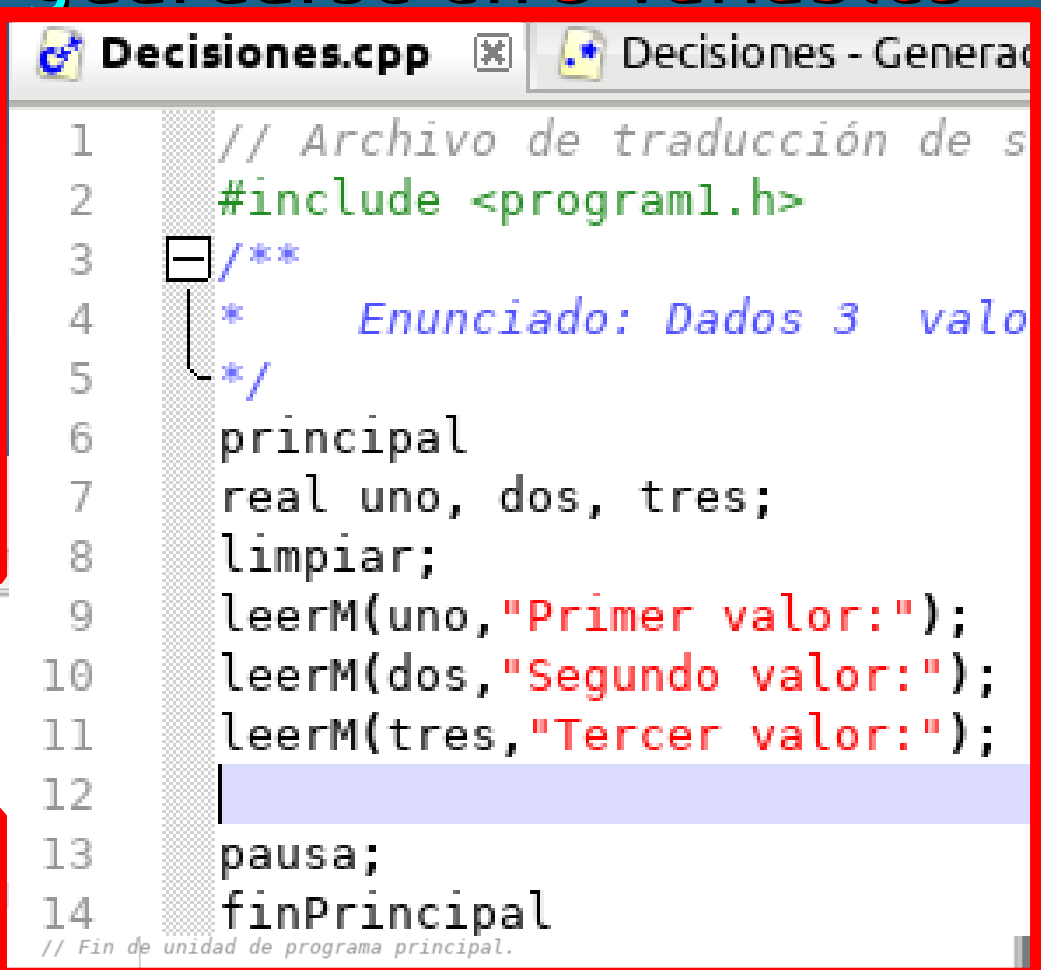
Dados 3 valores, mostrarlos de menor a mayor.

Los datos deben leerse y guardarse en 3 variables numéricas reales:

- `real uno, dos, tres;`



```
1 // Archivo de traducción de pseudocódigo a C++
2 #include <program1.h>
3 /**
4  * Enunciado: Dados 3 valores, mostrarlos de menor a mayor.
5  */
6 principal
7 real uno, dos, tres;
8 limpiar;
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12
13 pausa;
14 finPrincipal
```



```
1 // Archivo de traducción de s
2 #include <program1.h>
3 /**
4  * Enunciado: Dados 3 valo
5  */
6 principal
7 real uno, dos, tres;
8 limpiar;
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12
13 pausa;
14 finPrincipal
// Fin de unidad de programa principal.
```



Dados 3 valores, mostrarlos de menor a mayor.

Uno de los 3 valores es el menor:

Decisiones.cpp

```
1 // Archivo de traducción
2 #include <program1.h>
3 /**
4  * Enunciado: Dados 3
5  */
6 principal
7 real uno, dos, tres;
8 limpiar:
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces
13     sinoSi(dos <= tres) entonces
14         sino
15             finSi
16 pausa;
17 finPrincipal
```

```
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces
13     sinoSi(dos <= tres) entonces
14         sino
15             finSi
```

```
// Datos (a leer)
// Limpia la pantalla.
```

```
// El menor es uno,
// sino, si es dos
// sino, es tres.
```

```
// Pausa antes de finalizar.
// Fin de unidad de programa principal.
```



Dados 3 valores, mostrarlos de menor a mayor.

Uno de los 3 valores es el menor:

- Lo mostramos.
  - ▮ Quedan dos valores por mostrar.
  - ▮ Podemos resolver primero cualquiera de las 3 ramas:
    - Arbitrariamente empezamos por la 1ª.

```
12  si(unos <= dos Y unos <= tres) entonces           // El menor es uno,  
13      mostrar << unos << ", ";                     // y lo mostramos.  
14      sinoSi(dos <= tres) entonces                 // sino, si es dos  
15      mostrar << dos << ", ";                       // y lo mostramos.  
16      sino                                           // sino, es tres.  
17      mostrar << tres << ", ";                     // y lo mostramos.  
18  finSi
```



Dados 3 valores, mostrarlos de menor a mayor.

Resolvemos la **primer rama** de la **decisión anidada**:

- Si  $\text{dos} \leq \text{tres}$ , mostramos dos, tres.
- Sino ( $\neg C$ :  $\text{dos} > \text{tres}$ ) mostramos tres, dos.

Estamos en **condiciones de probar todos los casos** donde **el primero es el menor...**

```
12 si(unos <= dos Y unos <= tres) entonces // El menor es uno,
13     mostrar << unos << ", "; // y lo mostramos.
14     si(dos <= tres) entonces // Si dos <= tres, se completa con dos y tres:
15         mostrar << dos << " y " << tres << salto; // y los mostramos.
16         sino // Sino, es al revés: se completa con tres y dos.
17         mostrar << tres << " y " << dos << salto; // y los mostramos.
18     finSi
19 sinoSi(dos <= tres) entonces // sino, si es dos
20     mostrar << dos << ", "; // y lo mostramos.
21     sino // sino, es tres.
22     mostrar << tres << ", "; // y lo mostramos.
23 finSi
```



Dados 3 valores, mostrarlos de menor a mayor.

### ○ Prueba 1:

```
1 // Archivo de traducción de
2 #include <program1.h>
3 /**
4  *   Enunciado: Dados 3 val
5  */
6 principal
7 real uno, dos, tres;
8 limpiar;
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces
13     mostrar << uno << ", ";
14     si(dos <= tres) entonces
15         mostrar << dos << " y " << tres << salto;
16     sino
17         mostrar << tres << " y " << dos << salto;
18     finSi
19 sino si(dos <= tres) entonces
20     mostrar << dos << ", ";
21     sino
22     mostrar << tres << ", ";
23     finSi
24 pausa;
25 finPrincipal
```

Zinjal - Consola de Ejecucion

```
Primer valor:1
Segundo valor:2
Tercer valor:3
1, 2 y 3
En pausa. <Escape> para continuar...
```

```
// El menor es uno,
// y lo mostramos.
// Si dos <= tres, se completa con dos y tres:
// y los mostramos.
// Sino, es al revés: se completa con tres y dos.
// y los mostramos.

// sino, si es dos
// y lo mostramos.
// sino, es tres.
// y lo mostramos.

// Pausa antes de finalizar.
// Fin de unidad de programa principal.
```



Dados 3 valores, mostrarlos de menor a mayor.

### ○ Prueba 2:

```
1 // Archivo de traducción de
2 #include <program1.h>
3 /**
4  *   Enunciado: Dados 3 va
5  */
6 principal
7 real uno, dos, tres;
8 limpiar;
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces
13     mostrar << uno << ", ";
14     si(dos <= tres) entonces
15         mostrar << dos << " y " << tres << salto;
16     sino
17         mostrar << tres << " y " << dos << salto;
18     finSi
19 sinoSi(dos <= tres) entonces
20     mostrar << dos << ", ";
21     sino
22     mostrar << tres << ", ";
23     finSi
24 pausa;
25 finPrincipal
```

Zinjal - Consola de Ejecucion

```
Primer valor:1
Segundo valor:2
Tercer valor:2
1, 2 y 2
En pausa. <Escape> para continuar...
```

```
// El menor es uno,
// y lo mostramos.
// Si dos <= tres, se completa con dos y tres:
// y los mostramos.
// Sino, es al revés: se completa con tres y dos.
// y los mostramos.

// sino, si es dos
// y lo mostramos.
// sino, es tres.
// y lo mostramos.

// Pausa antes de finalizar.
// Fin de unidad de programa principal.
```



Dados 3 valores, mostrarlos de menor a mayor.

### ○ Prueba 3:



Zinjal - Consola de Ejecucion



Primer valor:1

Segundo valor:3

Tercer valor:2

1, 2 y 3

En pausa. <Escape> para continuar...

```
1 // Archivo de traducción de
2 #include <program1.h>
3 /**
4  *   Enunciado: Dados 3 va
5  */
6 principal
7 real uno, dos, tres;
8 limpiar;
9 leerM(uno, "Primer valor:");
10 leerM(dos, "Segundo valor:");
11 leerM(tres, "Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces
13     mostrar << uno << ", ";
14     si(dos <= tres) entonces
15         mostrar << dos << " y " << tres << salto;
16     sino
17         mostrar << tres << " y " << dos << salto;
18     finSi
19 sinoSi(dos <= tres) entonces
20     mostrar << dos << ", ";
21     sino
22     mostrar << tres << ", ";
23     finSi
24 pausa;
25 finPrincipal
```

```
// El menor es uno,
// y lo mostramos.
// Si dos <= tres, se completa con dos y tres:
// y los mostramos.
// Sino, es al revés: se completa con tres y dos.
// y los mostramos.

// sino, si es dos
// y lo mostramos.
// sino, es tres.
// y lo mostramos.

// Pausa antes de finalizar.
// Fin de unidad de programa principal.
```





Dados 3 valores, mostrarlos de menor a mayor.

- ¡Funciona! Ahora sólo queda **copiar** y **editar** las líneas 14-18 (*origen*) a las otras dos ramas:

```
1 // Archivo de traducción de pseudocódigo a C++
2 #include <program1.h>
3 /**
4  *   Enunciado: Dados 3 valores, mostrarlos de menor a mayor.
5  */
6 principal
7     real uno, dos, tres;
8     limpiar;
9     leerM(uno, "Primer valor:");
10    leerM(dos, "Segundo valor:");
11    leerM(tres, "Tercer valor:");
12    si(uno <= dos Y uno <= tres) entonces
13        mostrar << uno << ", ";
14        si(dos <= tres) entonces
15            mostrar << dos << " y " << tres << salto;
16            sino
17                mostrar << tres << " y " << dos << salto;
18            finSi
19        sinoSi(dos <= tres) entonces
20            mostrar << dos << ", ";
21            sino
22                mostrar << tres << ", ";
23            finSi
24    pausa;
25 finPrincipal
```

```
// Unidad de programa principal
// Datos (a leer)
// Limpia la pantalla.

// El menor es uno,
// y lo mostramos.
// Si dos <= tres, se completa con dos y tres:
// y los mostramos.
// Sino, es al revés: se completa con tres y dos.
// y los mostramos.

// sino, si es dos
// y lo mostramos.
// sino, es tres.
// y lo mostramos.

// Pausa antes de finalizar.
// Fin de unidad de programa principal.
```





Dados 3 valores, mostrarlos de menor a mayor.

- Segunda rama: se cambia dos por uno.

The screenshot shows a C++ IDE with a code editor on the left and a 'Reemplazar' (Replace) dialog box on the right. The code editor displays a C++ program for finding the minimum of three numbers. The 'Reemplazar' dialog box is open, showing the search text 'dos' and the replacement text 'uno'. The 'Buscar en' (Search in) dropdown is set to 'Archivo actual' (Current file). The 'Reemplazar' button is highlighted.

```
1 // Archivo de traducción de pseudocódigo a C++
2 #include <program1.h>
3 /**
4  *   Enunciado: Dados 3 valores, mostrarlos de menor a mayor.
5  */
6 principal
7     real uno, dos, tres;
8     limpiar;
9     leerM(uno, "Primer valor:");
10    leerM(dos, "Segundo valor:");
11    leerM(tres, "Tercer valor:");
12    si(uno <= dos Y uno <= tres) entonces
13        mostrar << uno << ", ";
14        si(dos <= tres) entonces
15            mostrar << dos << " y " << tres << salto;
16        sino
17            mostrar << tres << " y " << dos << salto;
18        finSi
19    sinoSi(dos <= tres) entonces
20        mostrar << dos << ", ";
21    si(uno <= tres) entonces
22        mostrar << uno << " y " << tres << salto;
23    sino
24        mostrar << tres << " y " << dos << salto;
25    finSi
```

Reemplazar

Texto a buscar: dos

Reemplazar por: uno

☐ Solo palabras completas

☐ Solo al comienzo de la palabra

☐ Distinguir mayúsculas y minúsculas

☐ Ignorar comentarios

☐ Es una expresión regular

Buscar en: Archivo actual

☐ Cerrar este diálogo después de encontrar

Reemplazar

Reemplazar Todo

Buscar Siguiente

Buscar Anterior

Cancelar

os y tres:

n tres y dos.

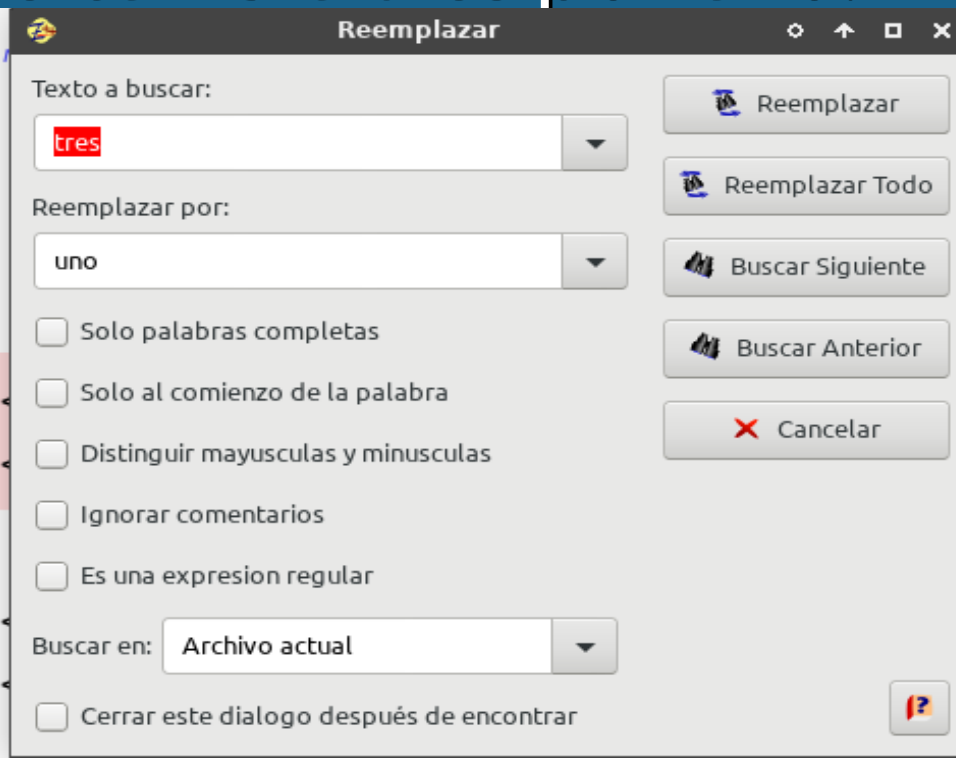
// Si uno <= tres, se completa con uno y tres:  
// y los mostramos.  
// Sino, es al revés: se completa con tres y uno.  
// y los mostramos.



Dados 3 valores, mostrarlos de menor a mayor.

○ Tercera rama: se cambia tres por uno.

```
3  /**
4   *   Enunciado: Dados 3 valores, mostrarlos de menor a mayor.
5   */
6   principal
7   real uno, dos, tres;
8   limpiar;
9   leerM(uno, "Primer valor:");
10  leerM(dos, "Segundo valor:");
11  leerM(tres, "Tercer valor:");
12  si(uno <= dos Y uno <= tres) entonces
13      mostrar << uno << ", ";
14      si(dos <= tres) entonces
15          mostrar << dos << " y " << tres
16          sino
17              mostrar << tres << " y " << dos
18          finSi
19      sinoSi(dos <= tres) entonces
20          mostrar << dos << ", ";
21      si(uno <= tres) entonces
22          mostrar << uno << " y " << tres
23          sino
24              mostrar << tres << " y " << uno
25          finSi
26      sino
27          mostrar << tres << ", ";
28      si(dos <= uno) entonces
29          mostrar << dos << " y " << uno << salto;
30      sino
31          mostrar << uno << " y " << dos << salto;
32      finSi
33  finSi
```



```
// y lo mostramos.
// Si dos <= uno, se completa con dos y uno:
// y los mostramos.
// Sino, es al revés: se completa con uno y dos.
// y los mostramos.
```



Listo.

```
#include <program1.h>
1  /*
2  *   Enunciado: Dados 3  valores, mostrarlos de menor a mayor.
3  */
4
5
6  principal                                // Unidad de programa principal
7  real uno, dos, tres;                    // Datos (a leer)
8  limpiar;                                // Limpia la pantalla.
9  leerM(uno,"Primer valor:");
10 leerM(dos,"Segundo valor:");
11 leerM(tres,"Tercer valor:");
12 si(uno <= dos Y uno <= tres) entonces    // El menor es uno,
13     mostrar << uno << ", ";              // y lo mostramos.
14     si(dos <= tres) entonces              // Si dos <= tres, se completa con dos y tres:
15         mostrar << dos << " y " << tres << salto; // y los mostramos.
16         sino                                // Sino, es al revés: se completa con tres y dos.
17         mostrar << tres << " y " << dos << salto; // y los mostramos.
18     finSi
19     sinoSi(dos <= tres) entonces          // sino, si es dos
20     mostrar << dos << ", ";              // y lo mostramos.
21     si(uno <= tres) entonces              // Si uno <= tres, se completa con uno y tres:
22         mostrar << uno << " y " << tres << salto; // y los mostramos.
23         sino                                // Sino, es al revés: se completa con tres y uno.
24         mostrar << tres << " y " << uno << salto; // y los mostramos.
25     finSi
26     sino                                // sino, es tres.
27     mostrar << tres << ", ";              // y lo mostramos.
28     si(dos <= uno) entonces              // Si dos <= uno, se completa con dos y uno:
29         mostrar << dos << " y " << uno << salto; // y los mostramos.
30         sino                                // Sino, es al revés: se completa con uno y dos.
31         mostrar << uno << " y " << dos << salto; // y los mostramos.
32     finSi
33     finSi
34 pausa;                                // Pausa antes de finalizar.
35 finPrincipal                            // Fin de unidad de programa principal.
```



Desarrolle dos programas para:

- Dado un año y un mes, *muestre la fecha del último día de ese mes y ese año.*
- Dado un año, un mes y un día, *calcule el número de días transcurridos desde el 31 de diciembre del año anterior.*



Desarrolle un programa para *encontrar las raíces de la siguiente ecuación:*

$$a.x^2 + b.x + c = 0$$