

ESTRUCTURAS DE REPETICION COMBINADAS

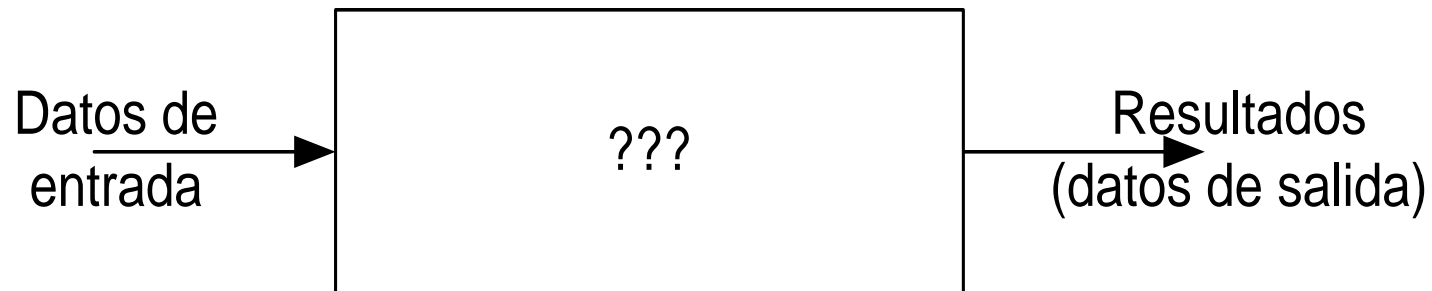
En el desarrollo de un programa no existen limitaciones para el uso de las estructuras de programación.

Se puede combinar todo con todo: secuencia, ciclos, decisiones, etc.

La selección de las estructuras a utilizar depende del programa que se quiera hacer, y en particular de la forma en la que se organizan los datos a procesar.

ESTRUCTURAS DE REPETICION COMBINADAS

- La clave está en analizar cuidadosamente el enunciado, y esto se consigue aplicando la metodología de análisis explicada, que plantea que sea cual sea la complejidad del problema a resolver podemos representarlo con el siguiente esquema:



ESTRUCTURAS DE REPETICION COMBINADAS

Ejemplo: dadas las edades de los alumnos de 10 cursos, calcular e informar la edad máxima de cada uno de ellos.

Nota: todos los cursos tienen 25 alumnos

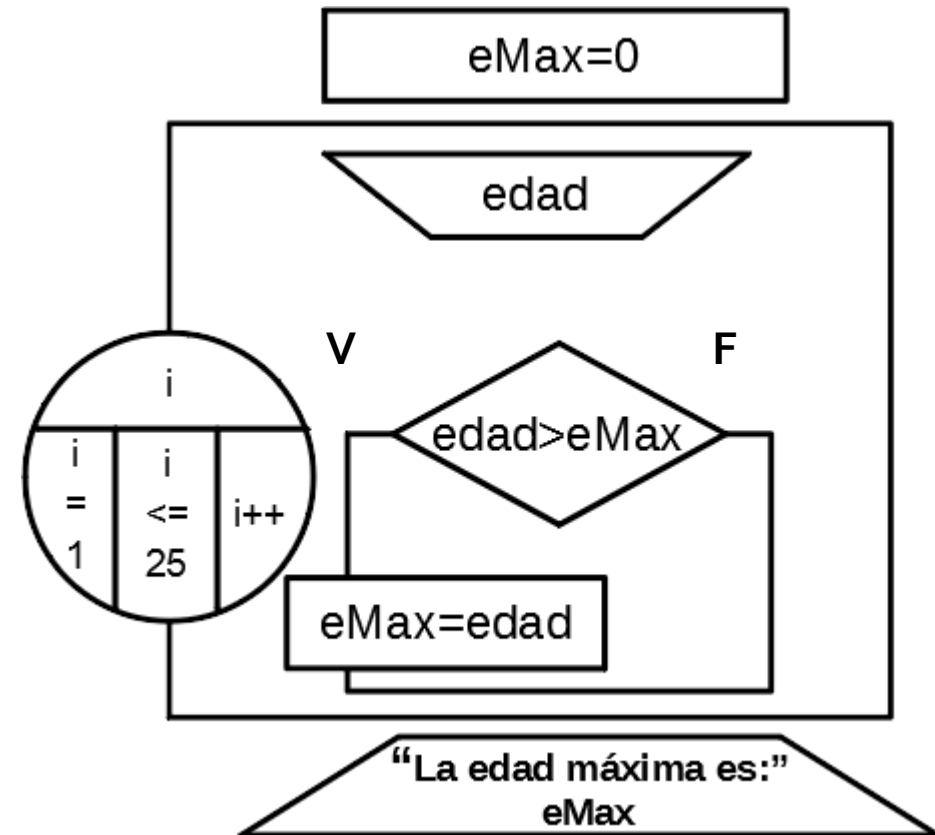
Datos de entrada: 10 lotes (1 para cada curso) de 25 valores de edad.

Datos de salida: 10 valores de edad máxima, 1 para cada curso.

ESTRUCTURAS DE REPETICION COMBINADAS

Por cada uno de los 10 lotes se tiene el siguiente algoritmo

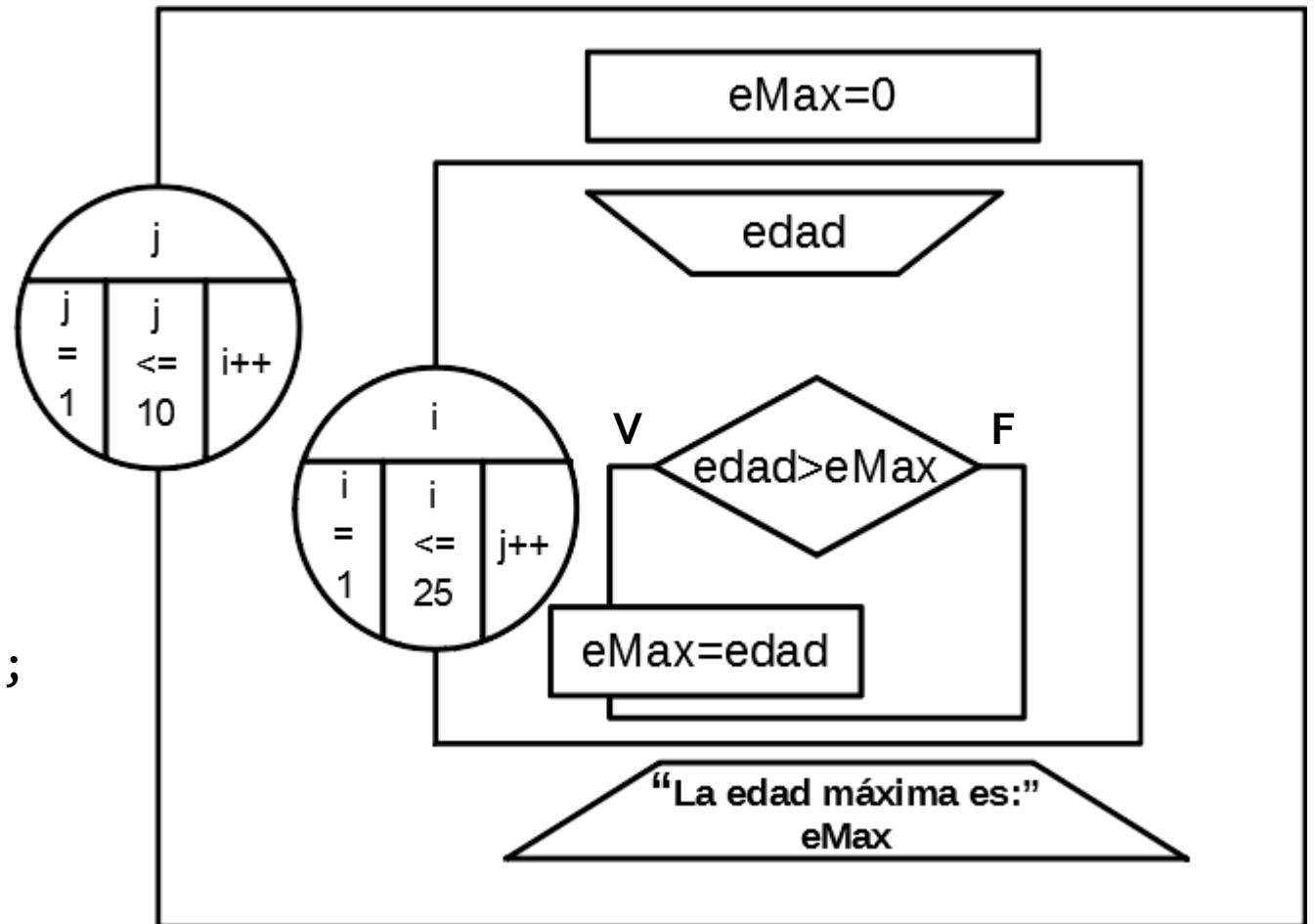
```
int main(){
    int i, edad, eMax=0;
    for(i=1;i<=25;i++){
        cout<<"INGRESE LA EDAD: ";
        cin>>edad;
        if(edad>eMax) eMax=edad;
    }
    cout<<"EDAD MAXIMA: "<<eMax<<endl;
    return 0;
}
```



ESTRUCTURAS DE REPETICION COMBINADAS

Para resolverlo se agrega otro ciclo: se repiten 10 veces las instrucciones.

```
int main(){
int i, edad, eMax, j;
for(j=1;j<=10;j++){
    eMax=0;
    for(i=1;i<=25;i++){
        cout<<"INGRESE LA EDAD: ";
        cin>>edad;
        if(edad>eMax) eMax=edad;
    }
    cout<<"EDAD MAXIMA: "<<eMax<<endl;
}
return 0;
}
```



ESTRUCTURAS DE REPETICION COMBINADAS

Ejemplo: dadas las edades de los alumnos de los cursos de TUP, calcular e informar la edad máxima de cada uno de ellos.

Nota: no se sabe la cantidad de cursos. Se ingresa primero el número de curso, y luego las edades de cada uno de los 25 integrantes del curso. Para indicar el fin de los cursos se ingresa un valor de curso igual a cero.

Datos de entrada: un número que indica el curso, y a continuación un conjunto de 25 valores de edad. Se repite una cantidad indeterminada de veces, hasta que el valor de curso sea cero.

Datos de salida: 1 valor de edad máxima, para cada curso. No se sabe cuantos.

Fuente: ciclo_comb_2.cpp

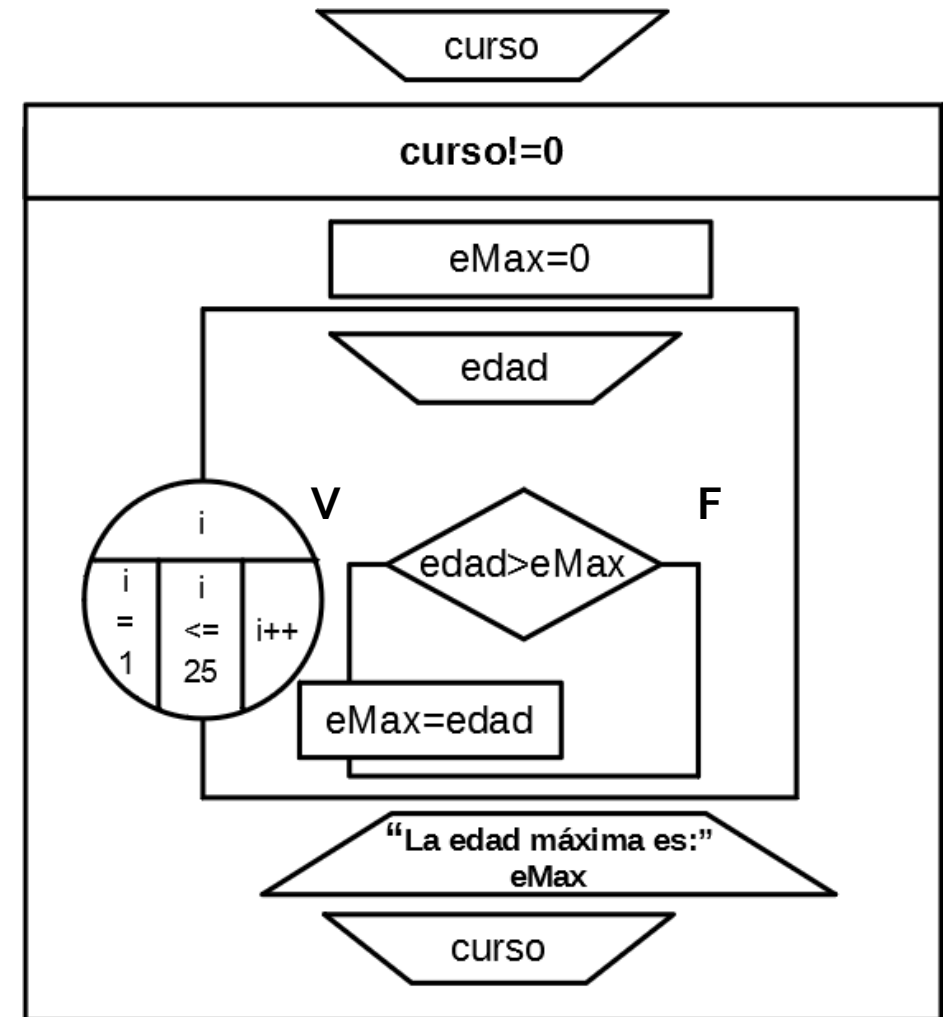
ESTRUCTURAS DE REPETICION COMBINADAS

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main(){
    int i, edad, eMax, j, curso;
    cout<<"CURSO: ";
    cin>>curso;
    while(curso!=0){
        eMax=0;
        for(i=1;i<=25;i++){
            cout<<"INGRESE LA EDAD: ";
            cin>>edad;
            if(edad>eMax) eMax=edad;
        }
        cout<<"EDAD MAXIMA: "<<eMax<<endl;
        cout<<"CURSO: ";
        cin>>curso;
    }
    return 0;
}
```

Fuente: ciclo_comb_2.cpp



ESTRUCTURAS DE REPETICION COMBINADAS

Ejemplo: dadas las edades de los alumnos de los 35 cursos de TUP, calcular e informar la edad máxima de cada uno de ellos.

Nota: no se sabe la cantidad de alumno de cada curso. Se ingresa primero el número de curso, y luego las edades de cada uno de los integrantes del curso. Para indicar el fin del curso se ingresa un valor de edad igual a cero.

Datos de entrada: un número que indica el curso, y a continuación un conjunto indeterminado de valores de edad que termina cuando se ingresa un 0. Se repite 35 veces.

Datos de salida: 1 valor de edad máxima, para cada uno de los 35 cursos.

Fuente: ciclo_comb_3.cpp

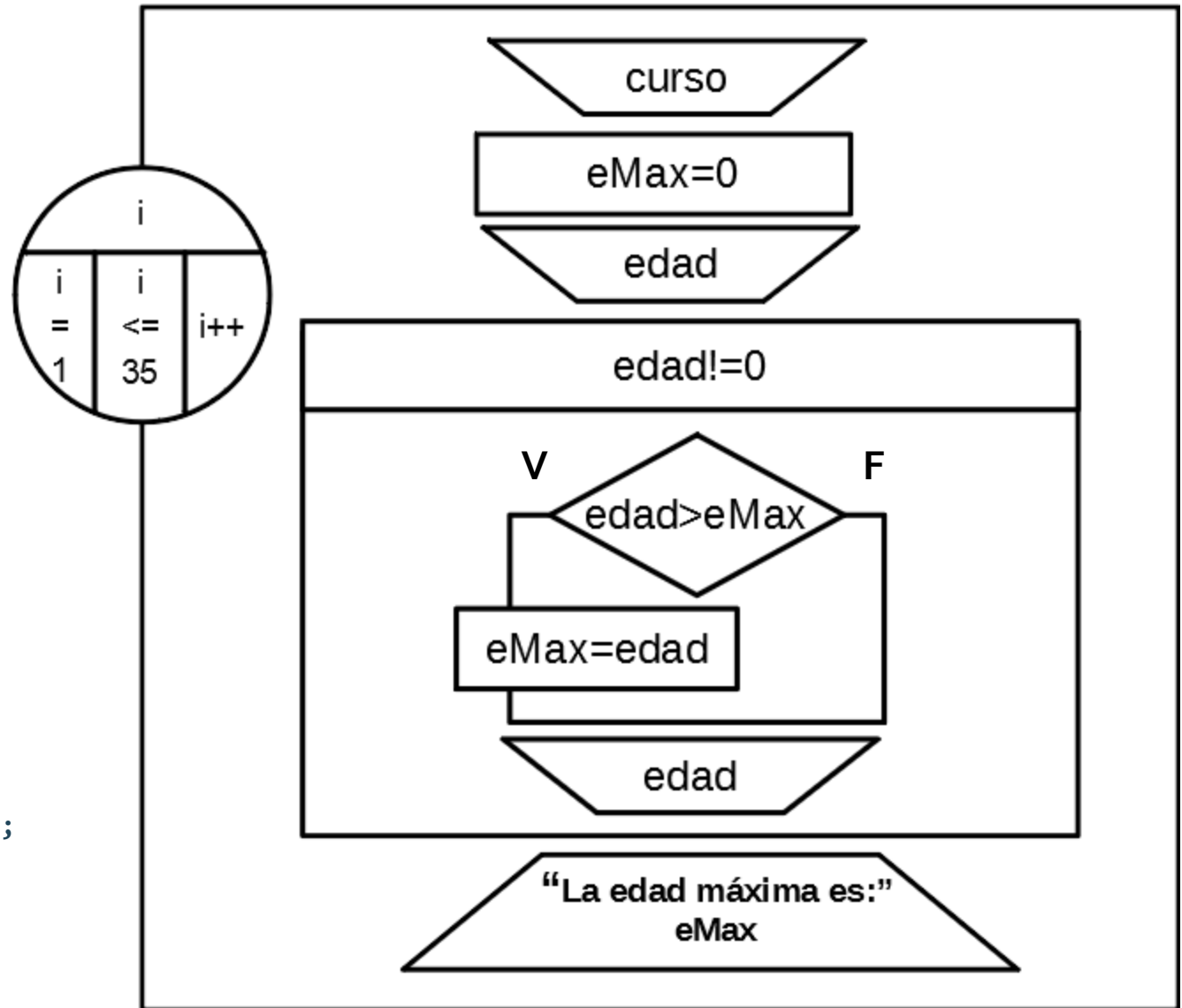
ESTRUCTURAS DE REPETICION COMBINADAS

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main(){
    int i, edad, eMax, curso;
    for(i=1;i<=35;i++){
        cout<<"CURSO: ";
        cin>>curso;
        eMax=0;
        cout<<"INGRESE LA EDAD: ";
        cin>>edad;
        while(edad!=0){
            if(edad>eMax) eMax=edad;
            cout<<"INGRESE LA EDAD: ";
            cin>>edad;
        }

        cout<<"EDAD MAXIMA: "<<eMax<<endl;
        cout<<"DEL CURSO: "<<curso<<endl<<endl;
    }
    return 0;
}
```



Fuente: ciclo_comb_3.cpp

ESTRUCTURAS DE REPETICION COMBINADAS

Ejemplo: dadas las edades de los alumnos de cada uno de los cursos de TUP, calcular e informar la edad máxima de cada uno de ellos.

Nota: no se sabe la cantidad de alumno de cada curso, ni la cantidad de cursos. Se ingresa primero el número de curso, y luego las edades de cada uno de los integrantes del curso. Para indicar el fin de cada curso se ingresa un valor de edad igual a cero. Para indicar el fin de los cursos se ingresa un número de curso negativo.

Datos de entrada: un número que indica el curso, y a continuación un conjunto indeterminado de valores de edad que termina cuando se ingresa un 0. Se repite una cantidad indeterminada de veces hasta que el número de curso que se ingresa sea negativo.

Datos de salida: 1 valor de edad máxima, para cada uno de los cursos.

Fuente: ciclo_comb_4.cpp

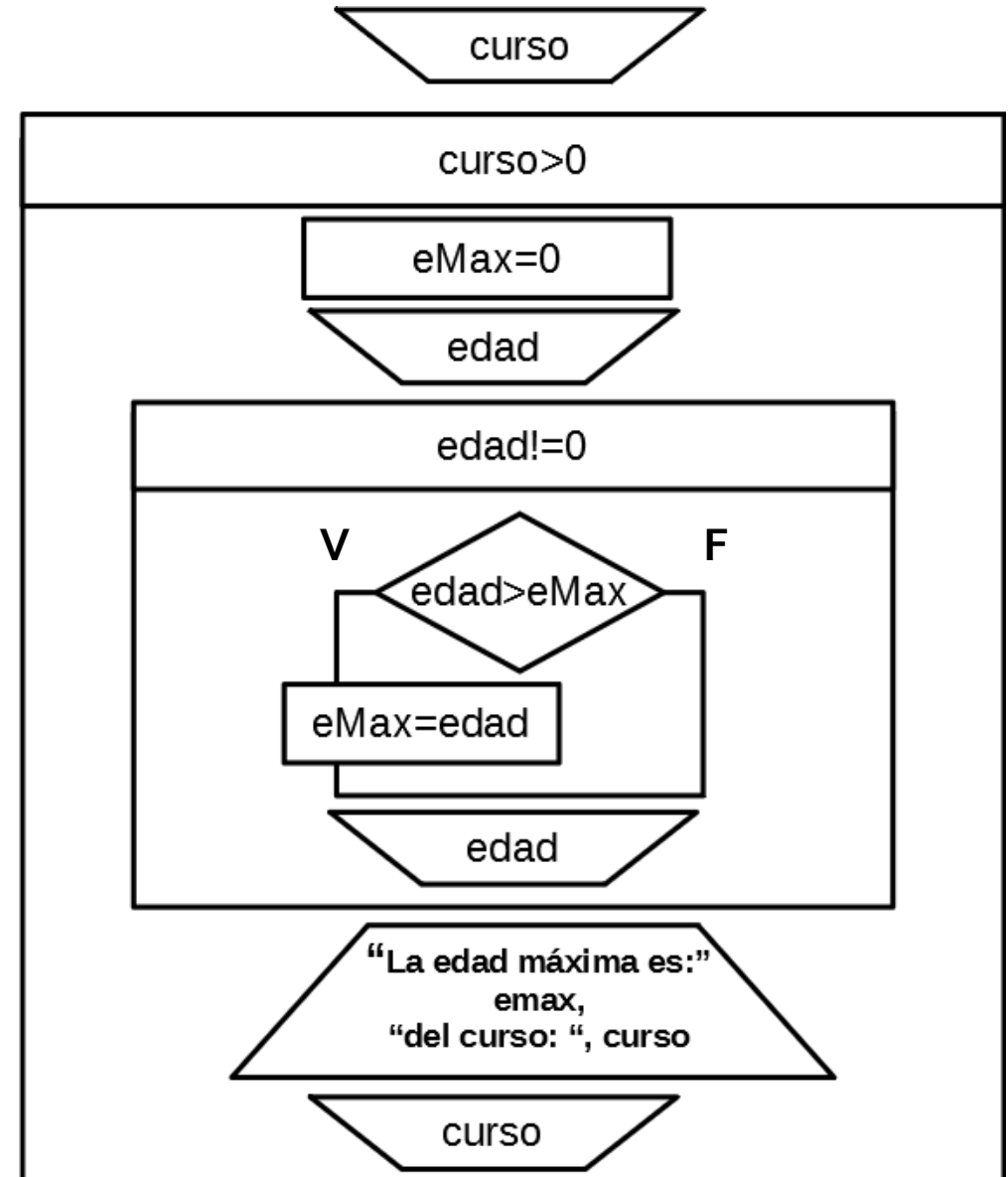
ESTRUCTURAS DE REPETICION COMBINADAS

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main(){
    int i, edad, eMax, j, curso;
    cout<<"CURSO: ";
    cin>>curso;
    while(curso>0){
        eMax=0;
        cout<<"INGRESE LA EDAD: ";
        cin>>edad;
        while(edad!=0){
            if(edad>eMax) eMax=edad;
            cout<<"INGRESE LA EDAD: ";
            cin>>edad;
        }
        cout<<"EDAD MAXIMA: "<<eMax<<endl;
        cout<<"DEL CURSO: "<<curso<<endl<<endl;
        cout<<"INGRESE NUEVO CURSO: ";
        cin>>curso;
    }
    return 0;
}
```

Fuente: ciclo_comb_4.cpp



ESTRUCTURAS DE REPETICION COMBINADAS

Ejemplo: dadas las edades de los alumnos de cada uno de los cursos de TUP, calcular e informar la edad máxima de cada uno de ellos.

Nota: no se sabe la cantidad de alumno de cada curso, ni la cantidad de cursos. Se tiene un conjunto de registros con el curso y la edad del alumno. Para indicar el fin de todos los cursos se ingresa un número de curso negativo.

Datos de entrada: un conjunto indeterminado de registros con el curso y la edad. Todos los datos de un curso están juntos (agrupados), por lo que se sabe que un curso se termina cuando se ingresa un valor de curso distinto al anterior. Se repite una cantidad indeterminada de veces, hasta que el número de curso sea negativo.

Datos de salida: 1 valor de edad máxima, para cada uno de los cursos.

Fuente: ciclo_comb_cc.cpp

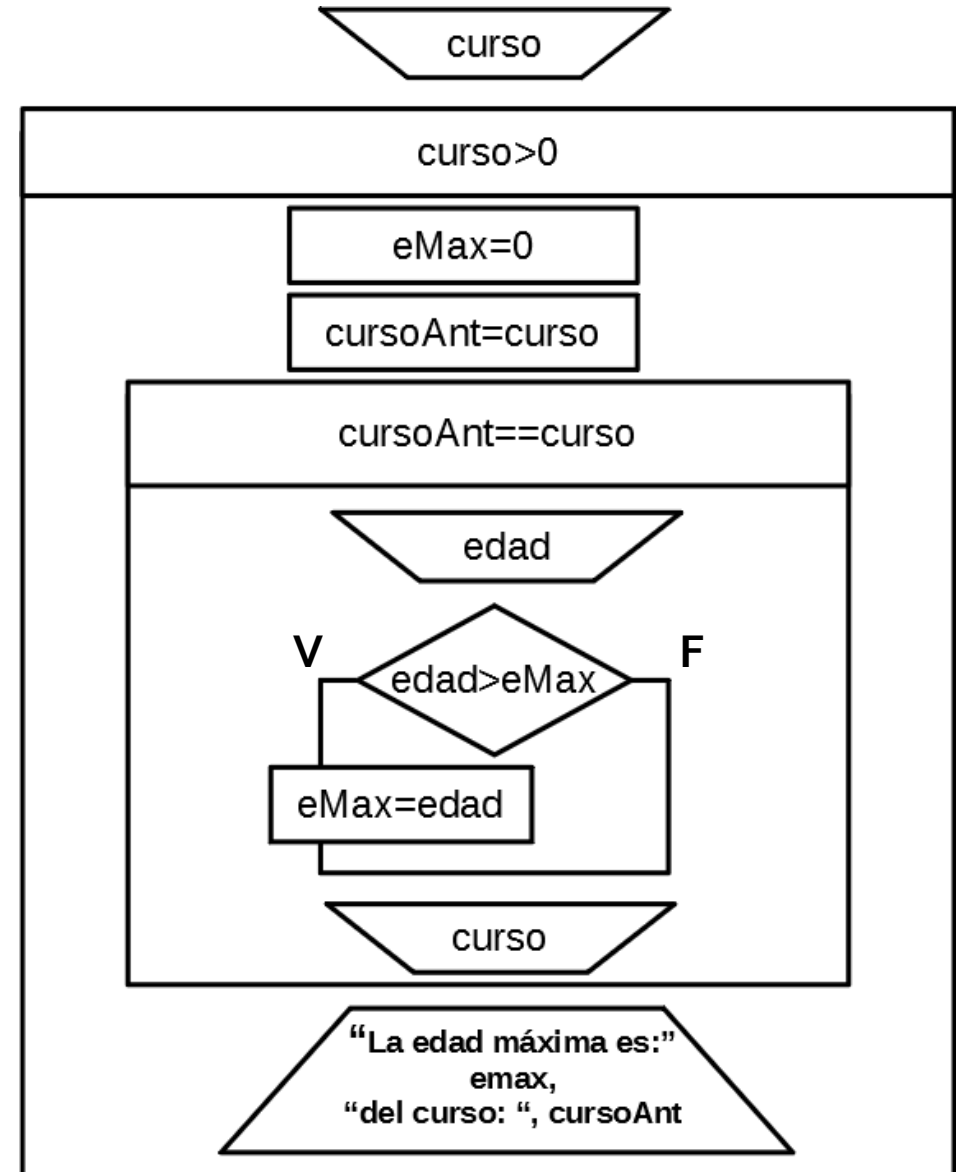
ESTRUCTURAS DE REPETICION COMBINADAS

```
#include <iostream>
#include <cstdlib>

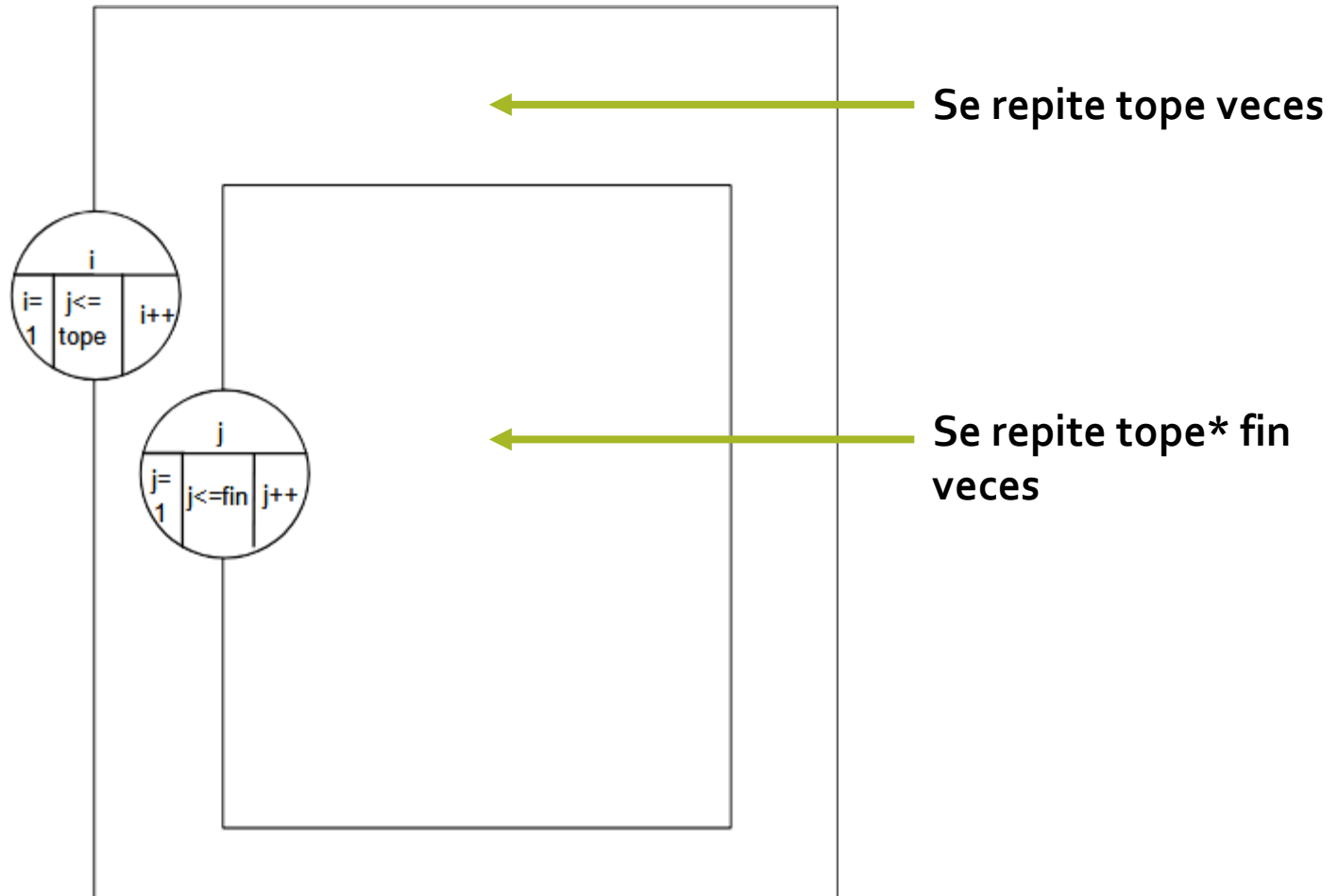
using namespace std;

int main(){
    int i, edad, eMax, j, curso, cursoAnt;
    cout<<"CURSO: ";
    cin>>curso;
    while(curso>0){
        eMax=0;
        cursoAnt=curso;
        while(curso==cursoAnt){
            cout<<"INGRESE LA EDAD: ";
            cin>>edad;
            if(edad>eMax) eMax=edad;
            cout<<"INGRESE CURSO: ";
            cin>>curso;
        }
        cout<<"EDAD MAXIMA: "<<eMax;
        cout<<" DEL CURSO: "<<cursoAnt<<endl<<endl;
        system("pause");
        system("cls");
    }
    return 0;
}
```

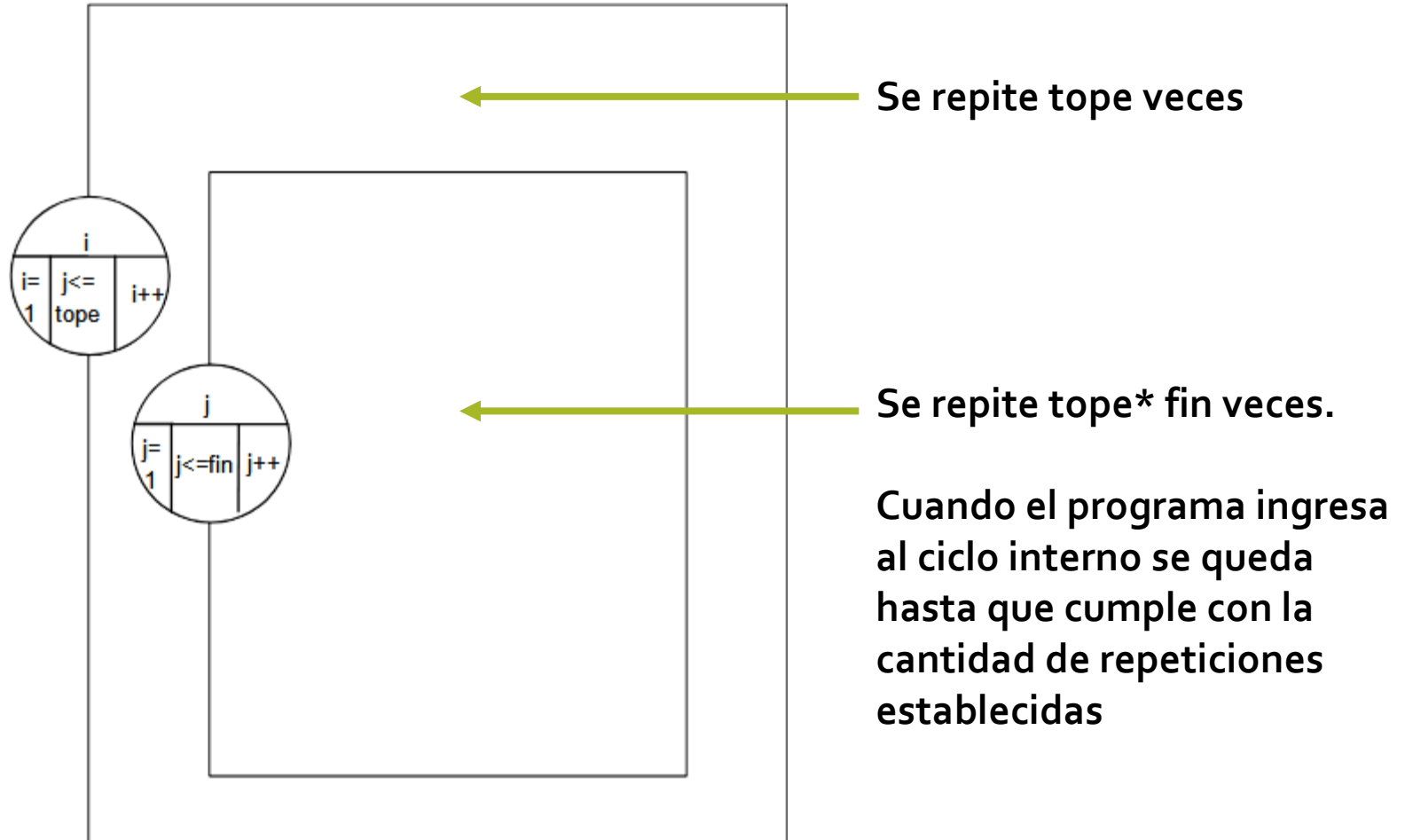
Fuente: ciclo_comb_cc.cpp



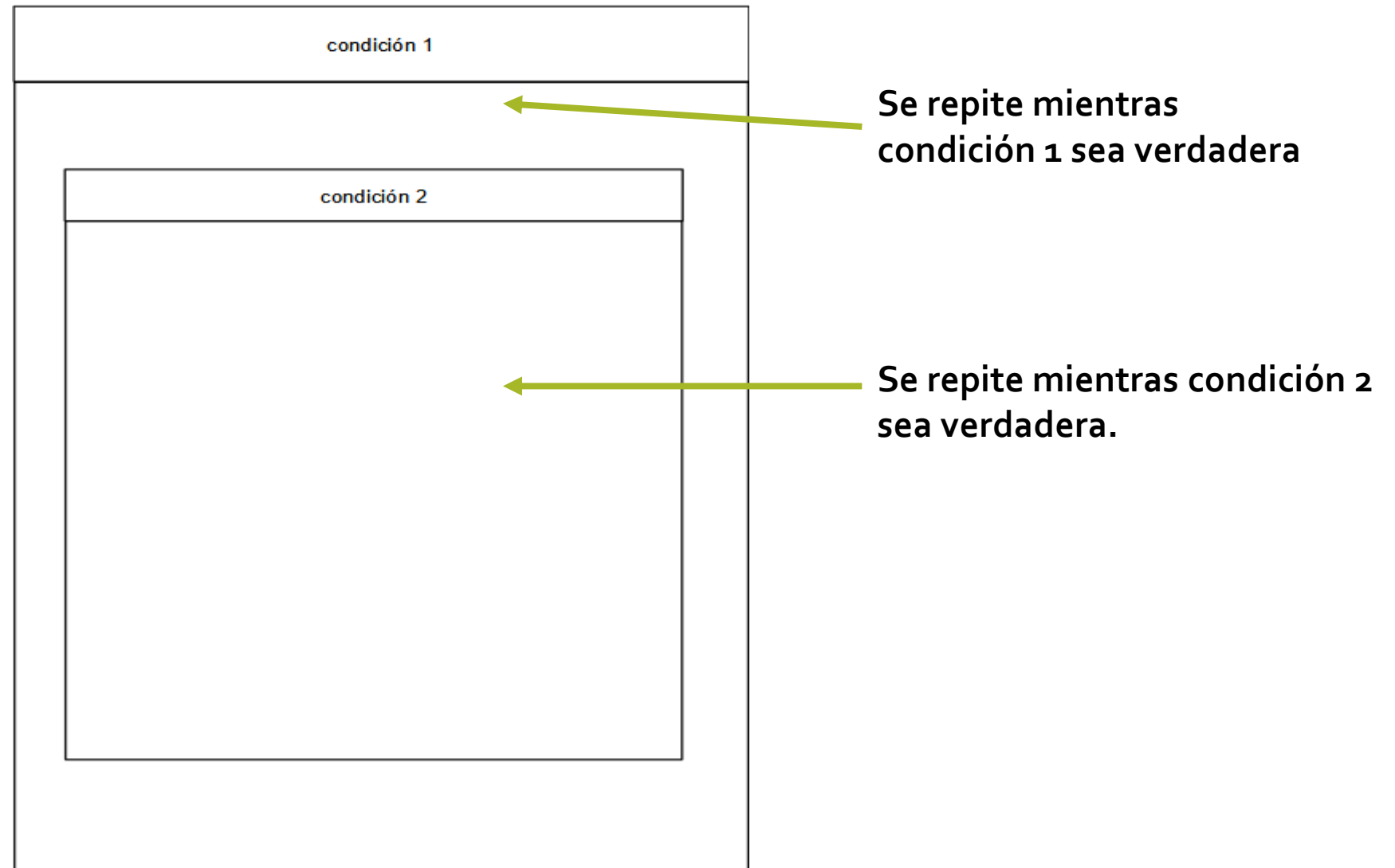
ESTRUCTURAS DE REPETICION COMBINADAS



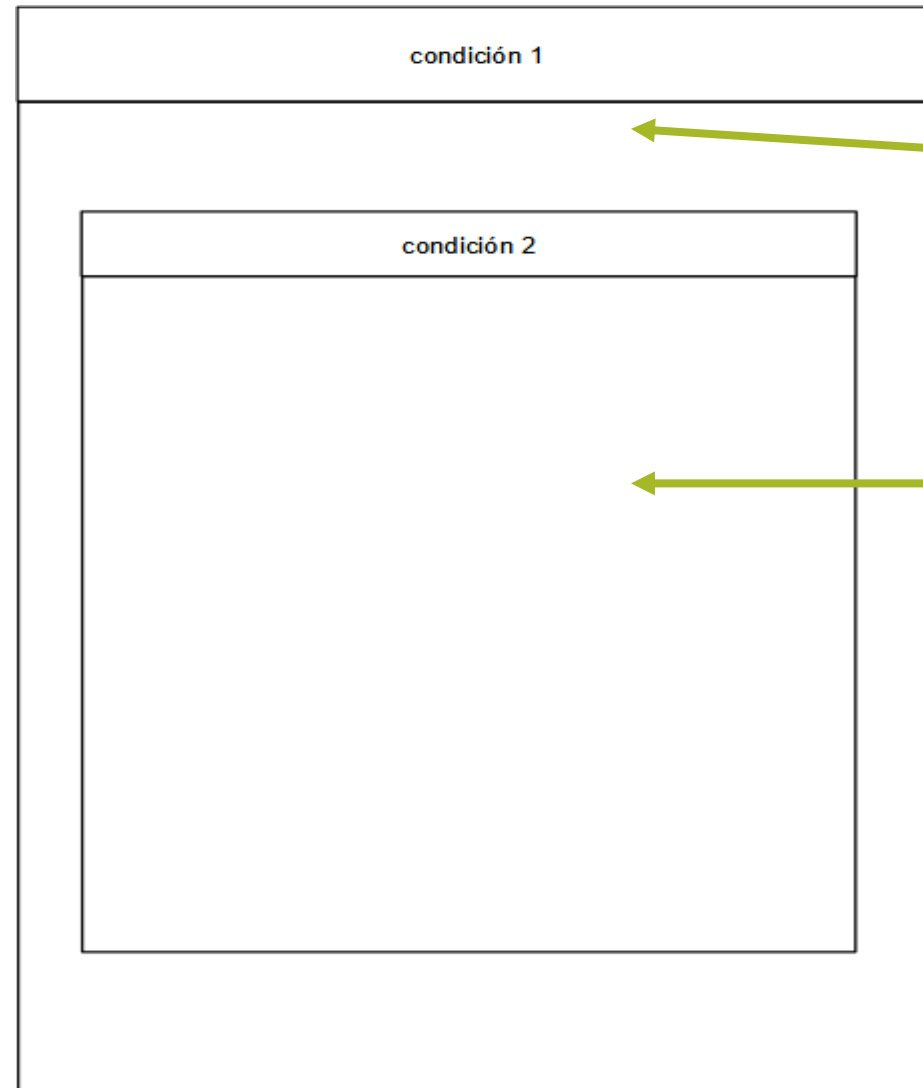
ESTRUCTURAS DE REPETICION COMBINADAS



ESTRUCTURAS DE REPETICION COMBINADAS



ESTRUCTURAS DE REPETICION COMBINADAS

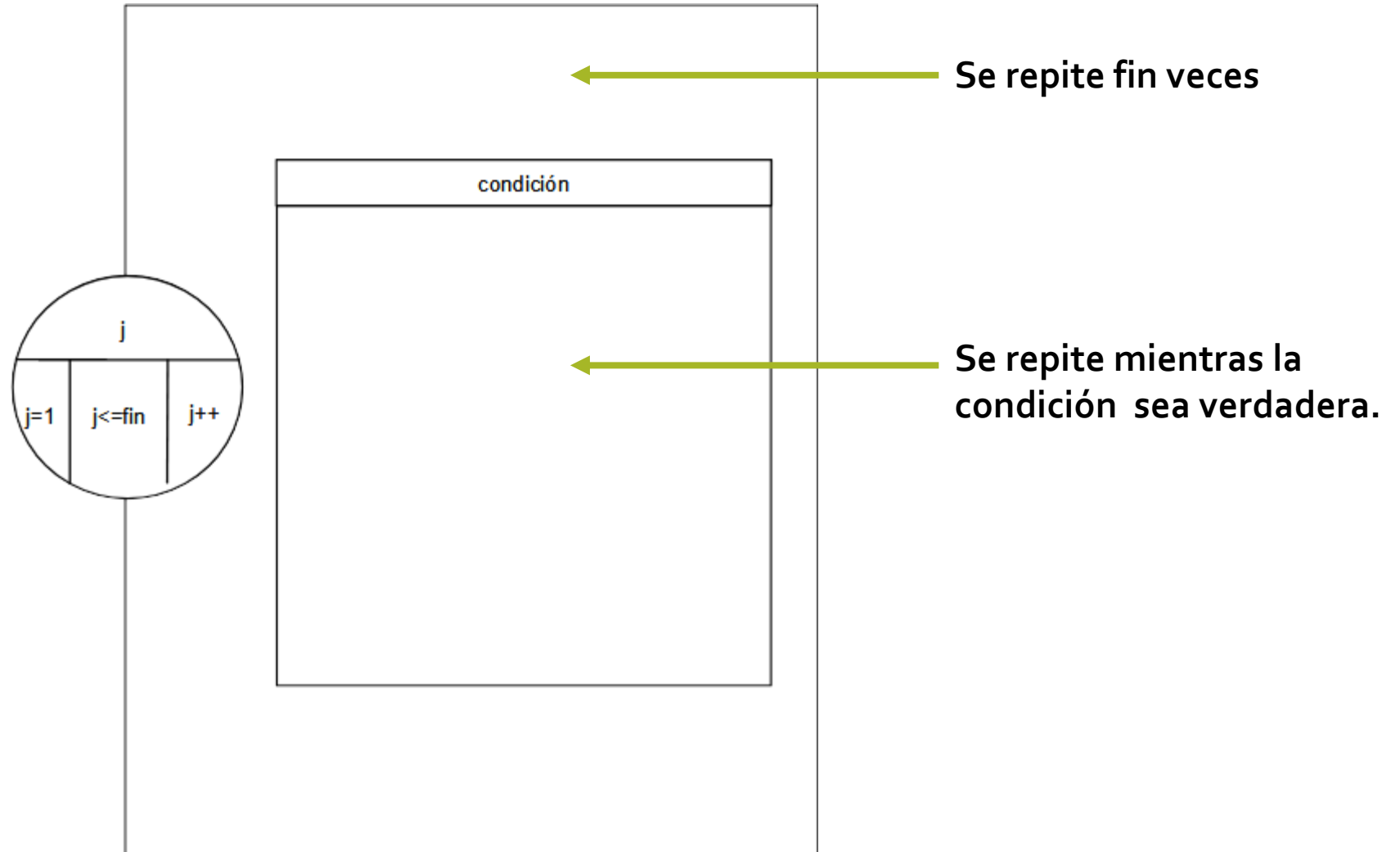


Se repite mientras
condición 1 sea verdadera

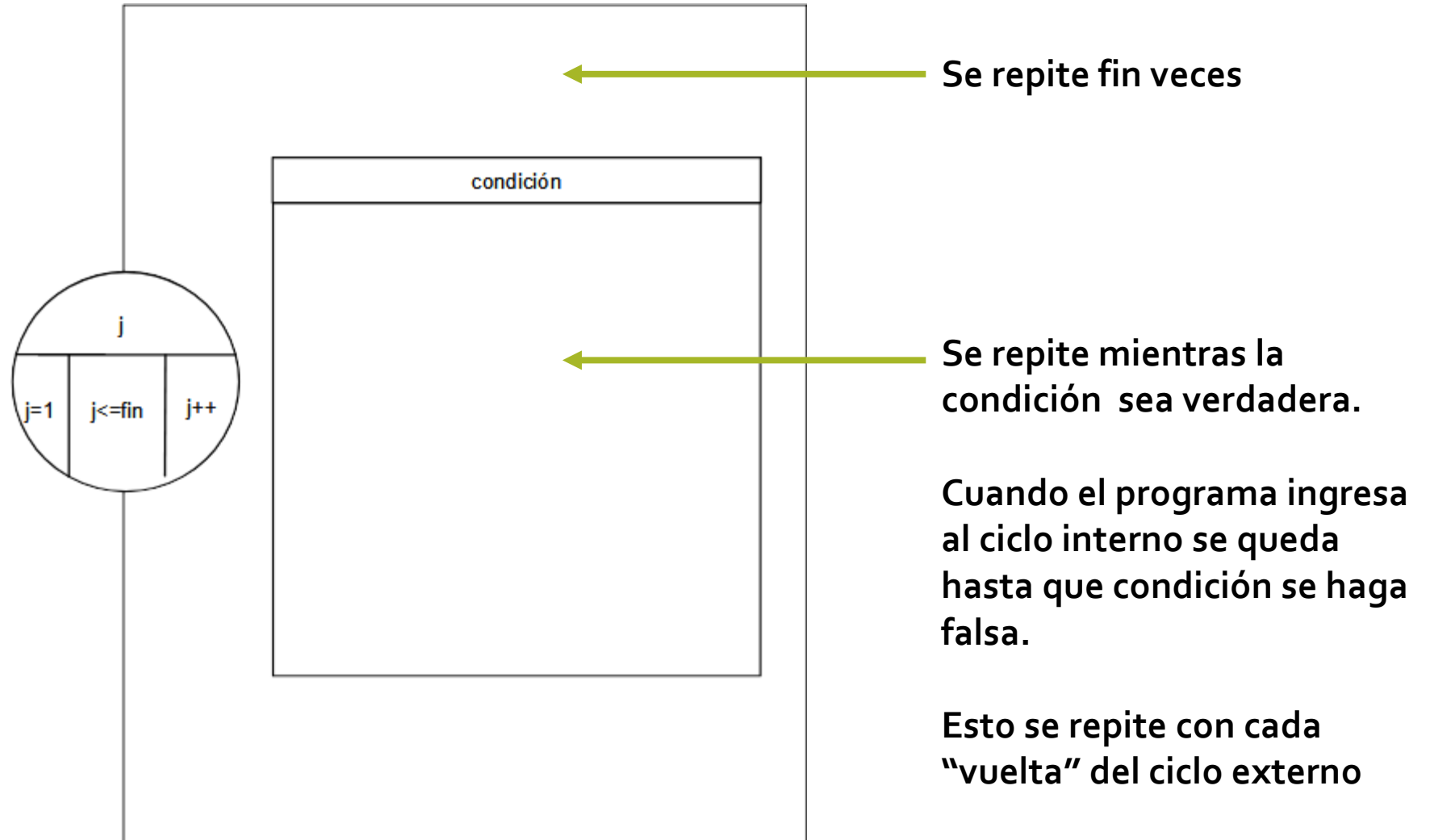
Se repite mientras condición
2 sea verdadera.

Cuando el programa ingresa
al ciclo interno se queda
hasta que condición 2 se
haga falsa.

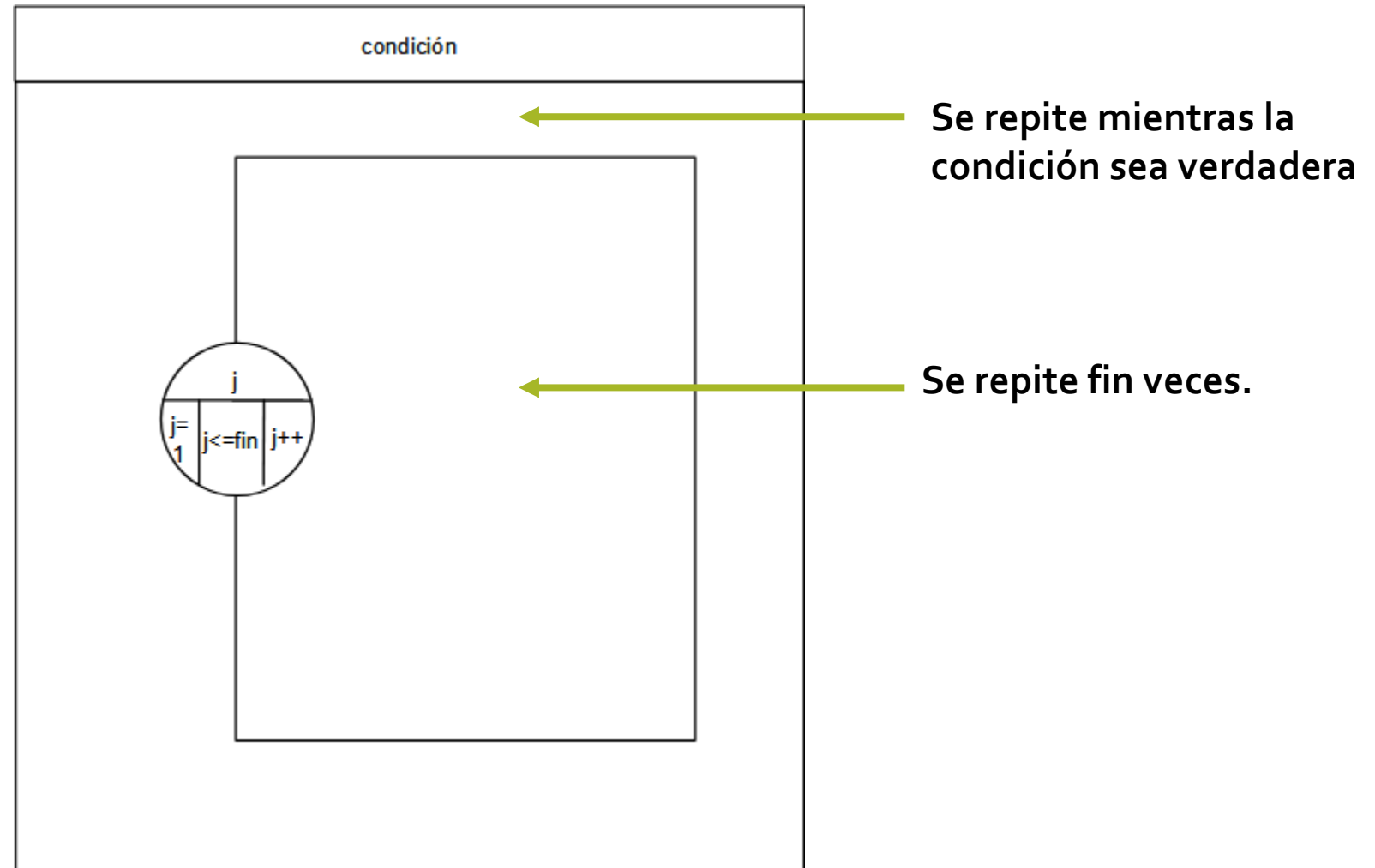
ESTRUCTURAS DE REPETICION COMBINADAS



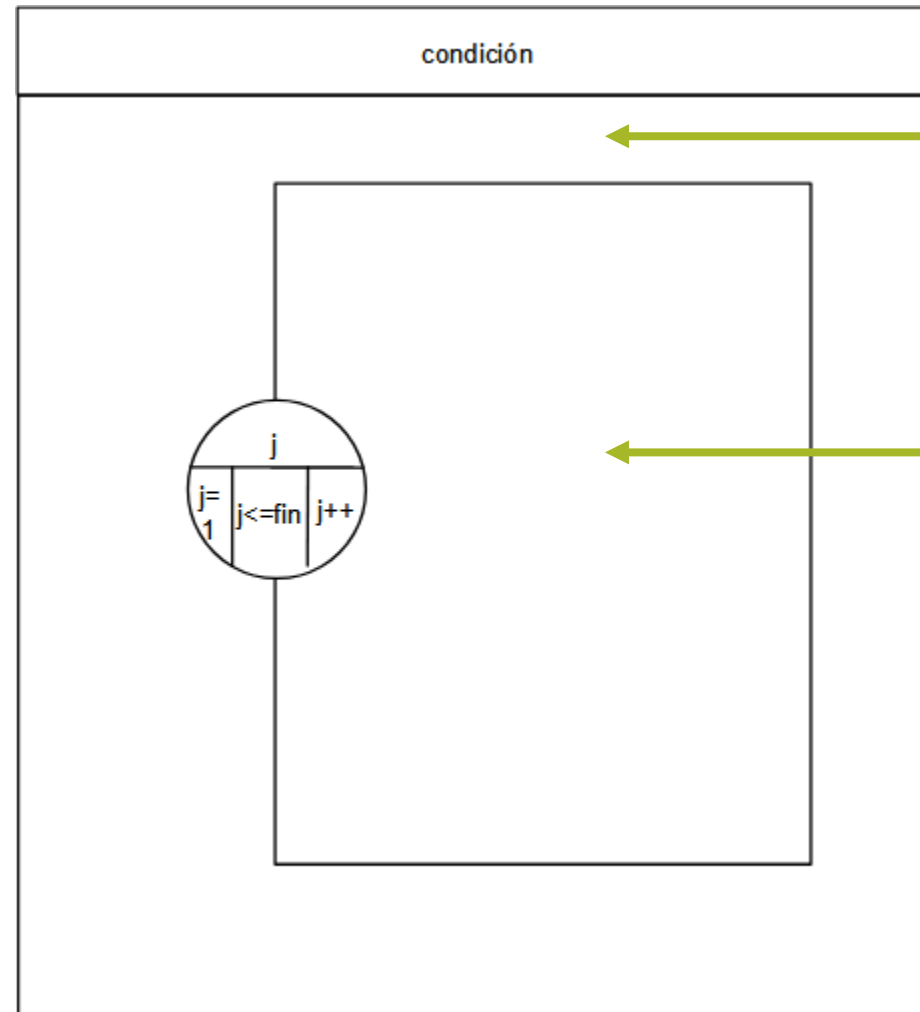
ESTRUCTURAS DE REPETICION COMBINADAS



ESTRUCTURAS DE REPETICION COMBINADAS



ESTRUCTURAS DE REPETICION COMBINADAS



Se repite mientras la condición sea verdadera

Se repite fin veces.

Cuando el programa ingresa al ciclo interno se queda hasta que se cumpla con las repeticiones establecidas.

Esto se vuelve a ejecutar con cada "vuelta" del ciclo externo