



aprenderaprogramar.com

Control sobre el flujo de programas. Instrucción finalizar. Pseudocódigo y diagramas de flujo. (CU00178A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 77 del Curso Bases de la programación Nivel I

24

INSTRUCCIÓN FINALIZAR

La instrucción *Finalizar* da pie a que se termine el procesamiento del algoritmo tal y como lo hace la instrucción *Fin*. Reservaremos *Fin* como palabra clave a disponer como última instrucción en la escritura de un algoritmo y *Finalizar* para cualquier terminación del programa que no coincida con el *Fin*.

Finalizar puede encontrarse en cualquier punto del programa comprendido entre *Inicio* y *Fin*, incluido una situación dentro de un bucle, un proceso de decisiones múltiples, etc. Normalmente *Finalizar* irá después de evaluar una situación que es la desencadenante de la finalización.

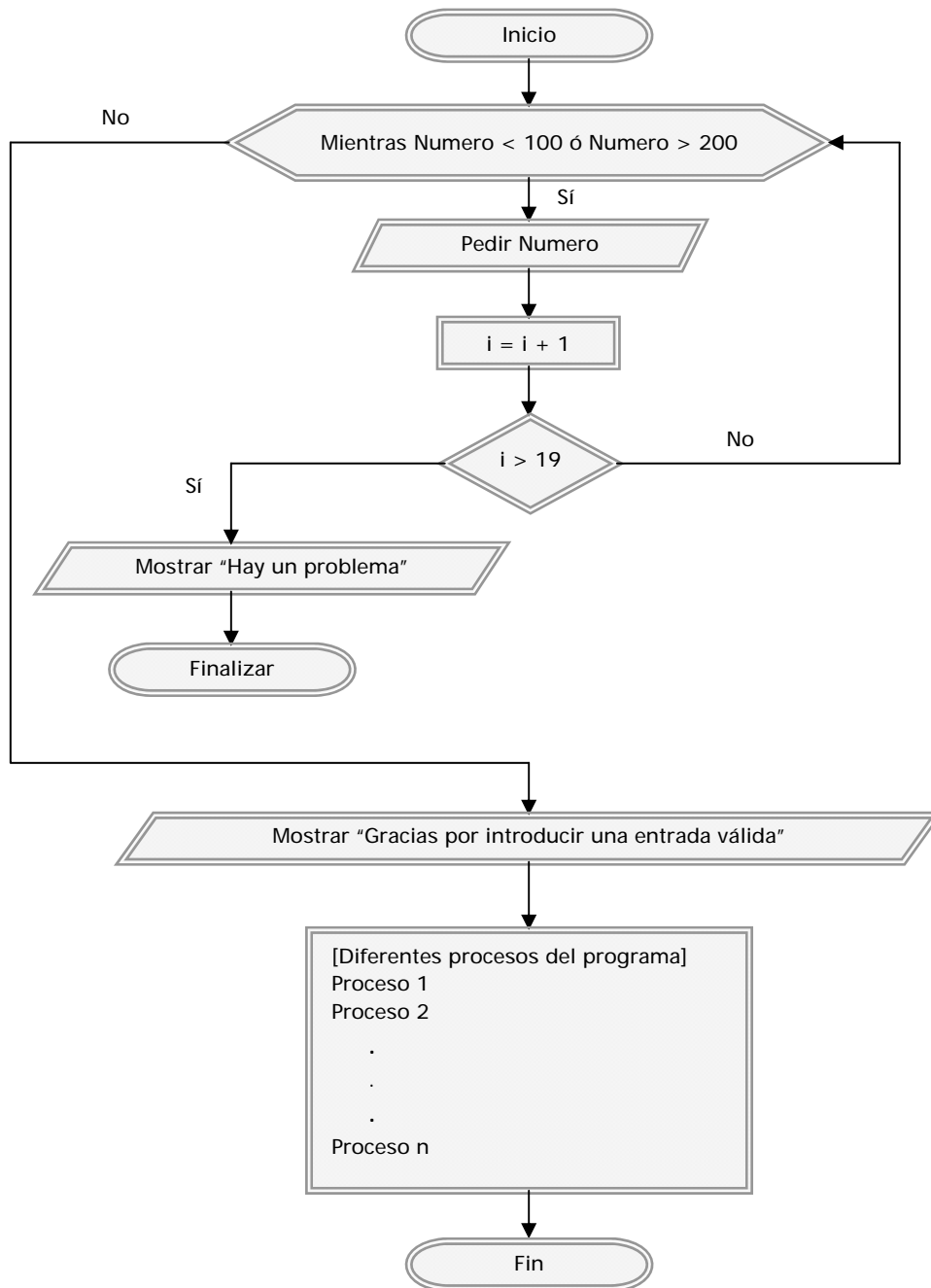
Equivalentes a *Finalizar* son *Acabar*, *Terminar* y los ampliamente difundidos términos anglosajones *End* y *Exit*.

Ejemplo de uso en un bucle.

Pseudocódigo:

```
1. Inicio [Finaliza si hay problemas – Ejemplo http://www.aprenderaprogramar.com]  
  
2. Mientras Numero < 100 ó Numero > 200 Hacer  
  
    Mostrar "Por favor, introduzca un número comprendido entre 100 y 200"  
  
    Pedir Numero  
  
    i = i + 1  
  
    Si i > 19 Entonces  
  
        Mostrar "Hay un problema para continuar y el programa se va a cerrar. Motivo:  
        20 procesos nulos. Consulte el manual del programa"  
  
        Finalizar  
  
    FinSi  
  
Repetir  
  
3. Mostrar "Gracias por introducir una entrada válida"  
  
4. [Aquí continuaría el programa con diferentes procesos]  
  
5. Fin
```

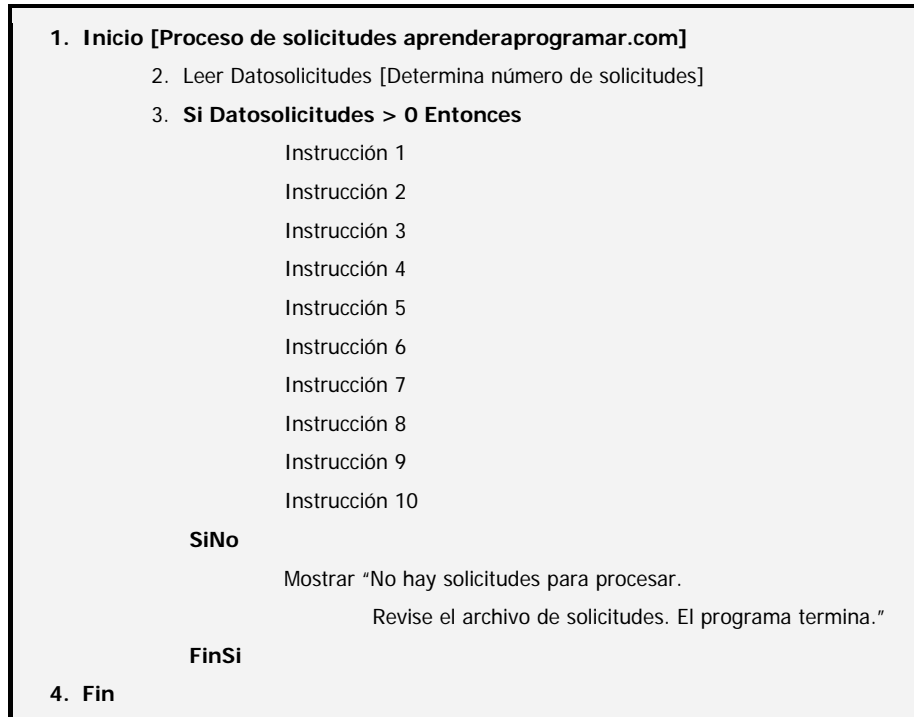
Diagrama de flujo:



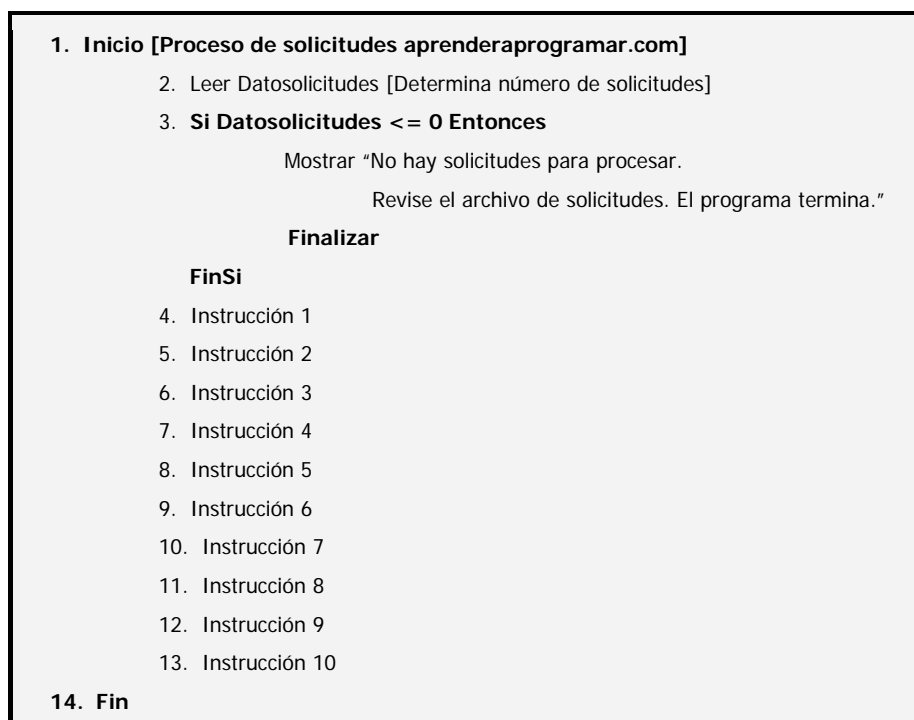
Comentarios: Es quizás la fuerte variación de la configuración del diagrama de flujo respecto a los que habitualmente veníamos haciendo, la que nos da una idea mejor de qué puede suponer la modificación directa del flujo de los programas. En este caso, se pide un número acotado en un intervalo y si el valor introducido no es válido se repite la petición. Si tras 20 intentos el valor no es válido se considera que existe un problema y se cierra el programa.

Ejemplo de uso fuera de un bucle.

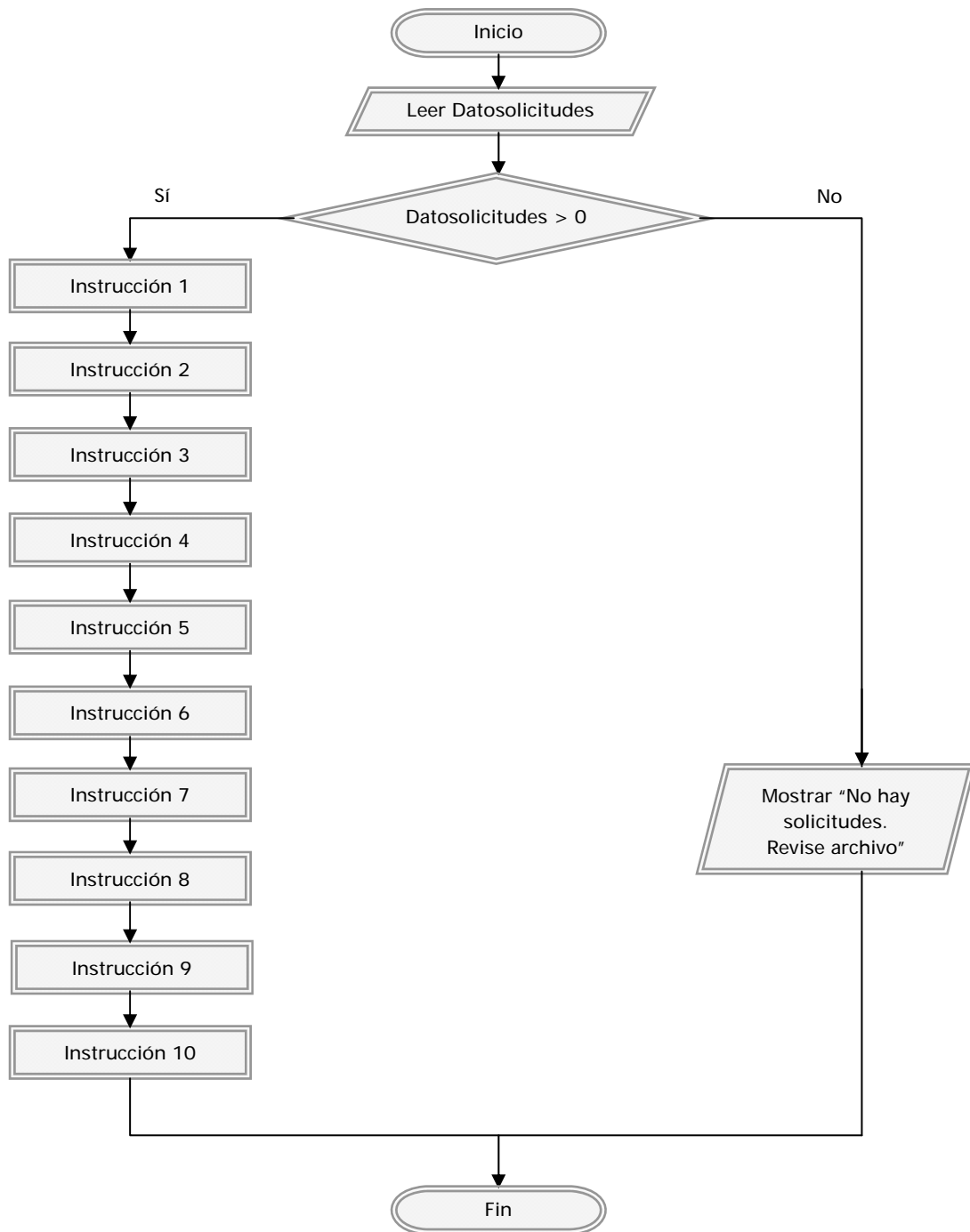
Consideremos un programa tal como este:



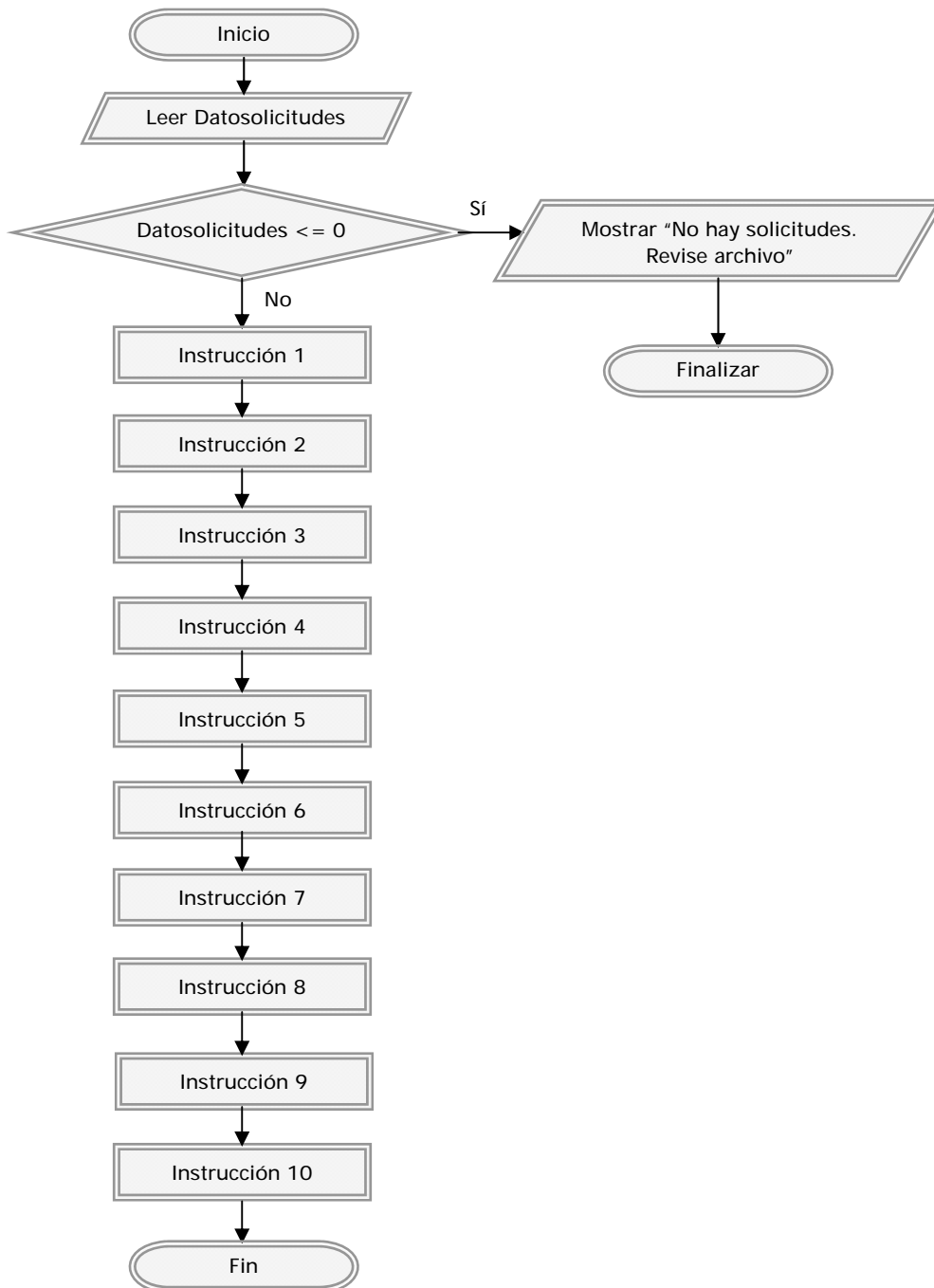
Y comparémoslo con este otro:



El diagrama de flujo para el primer caso sería:



Y para el segundo:



Comparando los pseudocódigos, vemos que en el primer caso el núcleo del programa, es decir, las instrucciones 1 a 10, quedan como bloque subordinado dentro de la instrucción *Sí*. Que todo el núcleo de un programa quede “sangrado” no es demasiado deseable. Analizando los diagramas de flujo vemos que en el primer caso hay dos ramas y en el segundo un tronco con un apéndice. Supongamos un programa no con 10 ó 15 instrucciones, sino con 1000 ó 1500 ¿Qué estructura aporta mayor claridad? ¿Cuál es más fácil de leer? ¿Cuál es más fácil de comprender?

No podemos decir “Nunca se dejará un núcleo de programa como bloque subordinado” ni “Nunca se usará un *Finalizar* para evitar un núcleo de programa subordinado”. Para cada caso particular el programador tendrá que estimar qué es lo mejor para su programa en función de su experiencia y criterio. De cualquier forma y a modo de conclusión:

- Evitar el uso de *Finalizar* siempre que sea posible.
- Recordar lo expuesto en la introducción a la modificación directa del flujo de programas.

Próxima entrega: CU00179A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59