



aprenderaprogramar.com

Instrucciones Si ... Entonces y Si ... Entonces – SiNo (if ... then ... else) en pseudocódigo (CU00142A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

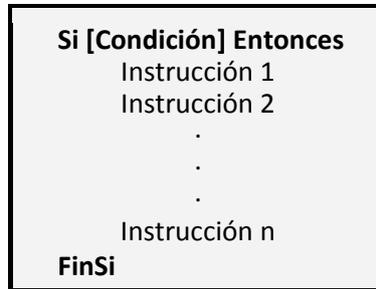
Autor: Mario R. Rancel

Resumen: Entrega nº 41 del Curso Bases de la programación Nivel I

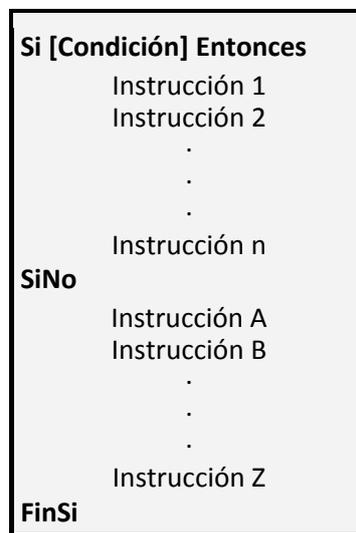
24

INSTRUCCIÓN SI ... ENTONCES Y SI ... ENTONCES – SINO

La instrucción *Si ... Entonces* permite controlar qué procesos tienen lugar en función del valor de una o varias variables o de las decisiones del usuario. Escribiremos esta instrucción de la siguiente manera:

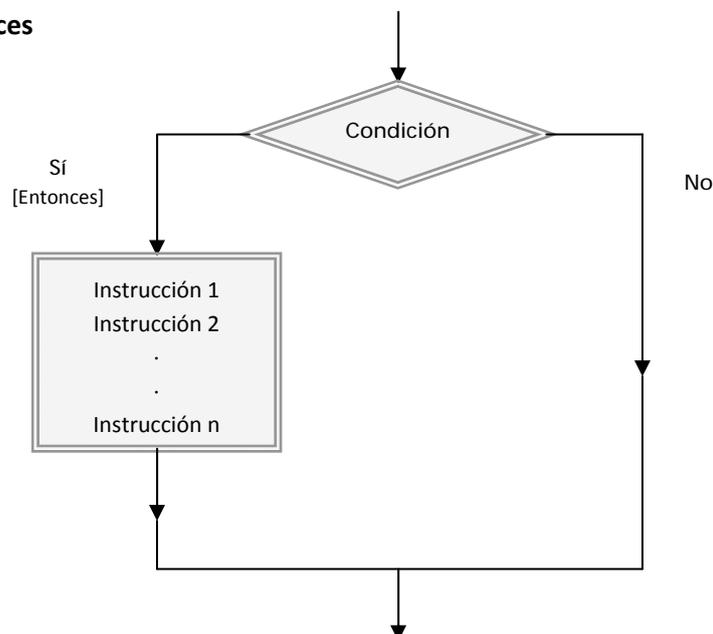


Se podrán introducir instrucciones a realizarse en caso de no cumplirse la condición:

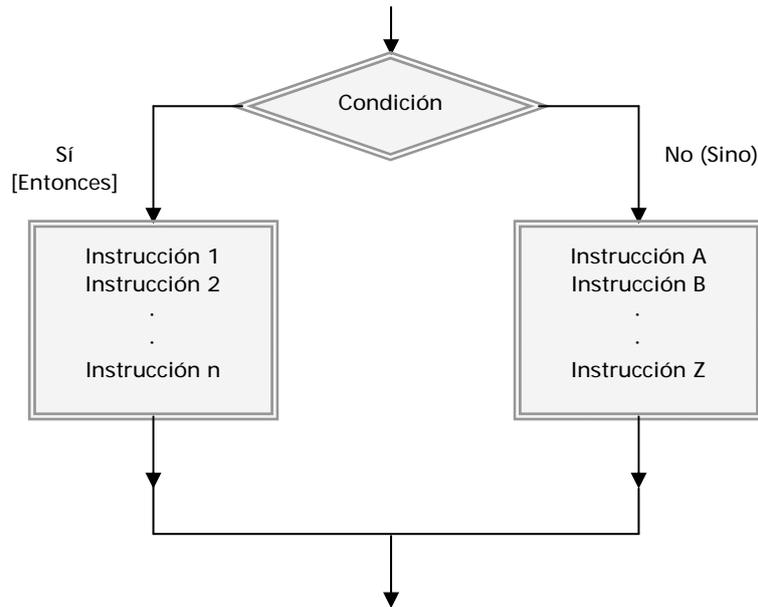


Nótese que el uso de sangrías permite identificar el bloque de sentencias a ejecutar. Gráficamente:

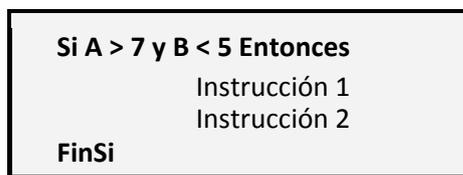
Instrucción Si ... Entonces



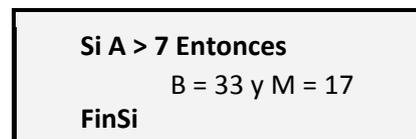
Instrucción Si ... Entonces - SiNo



Recordemos que para expresar condiciones disponemos de símbolos matemáticos como =, >, <, etc. así como de los operadores *y* (and) y *ó* (or).



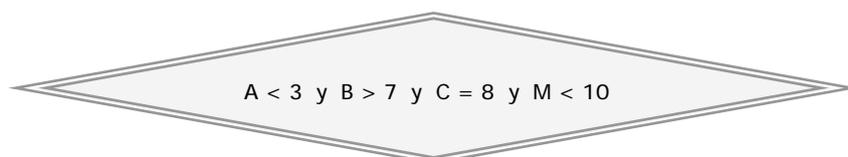
Válido



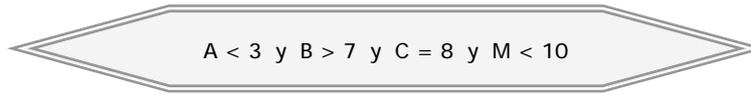
No válido (operador *y* no puede usarse como nexo entre instrucciones después de un entonces).

En cuanto a los diagramas de flujo, si una condición compuesta es de gran longitud, consideraremos válido achatar el rombo para adecuarlo al texto.

EJEMPLO



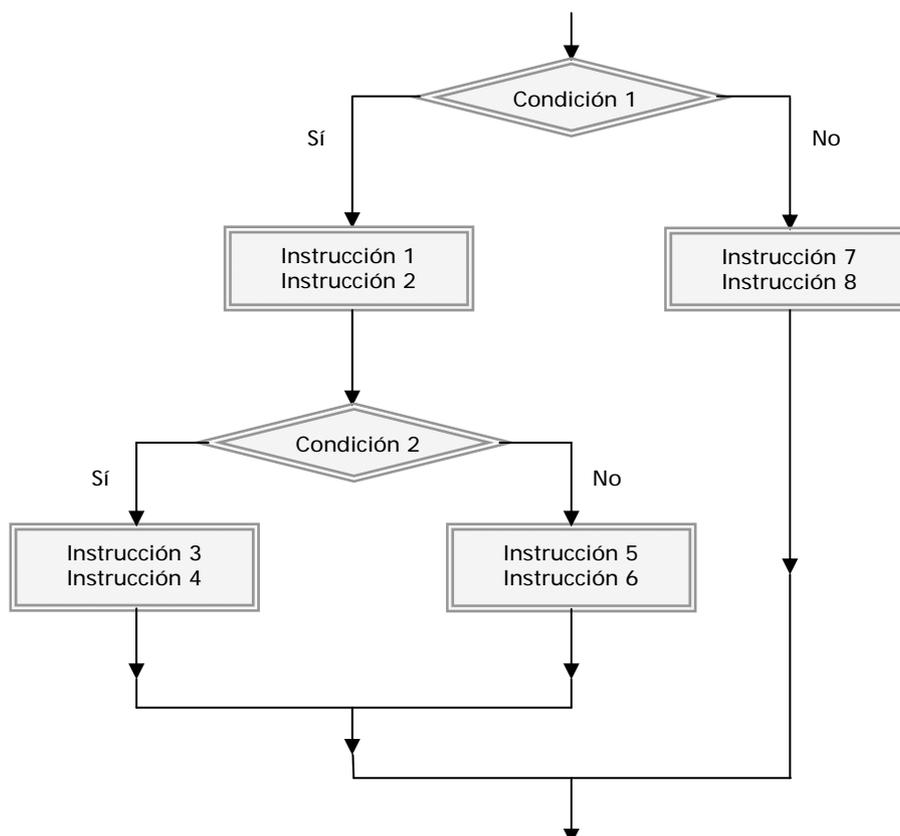
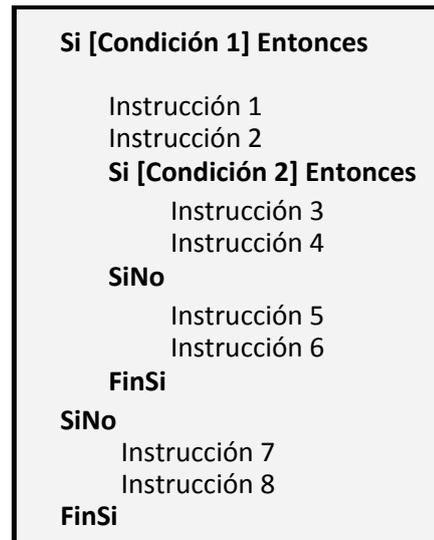
Válido



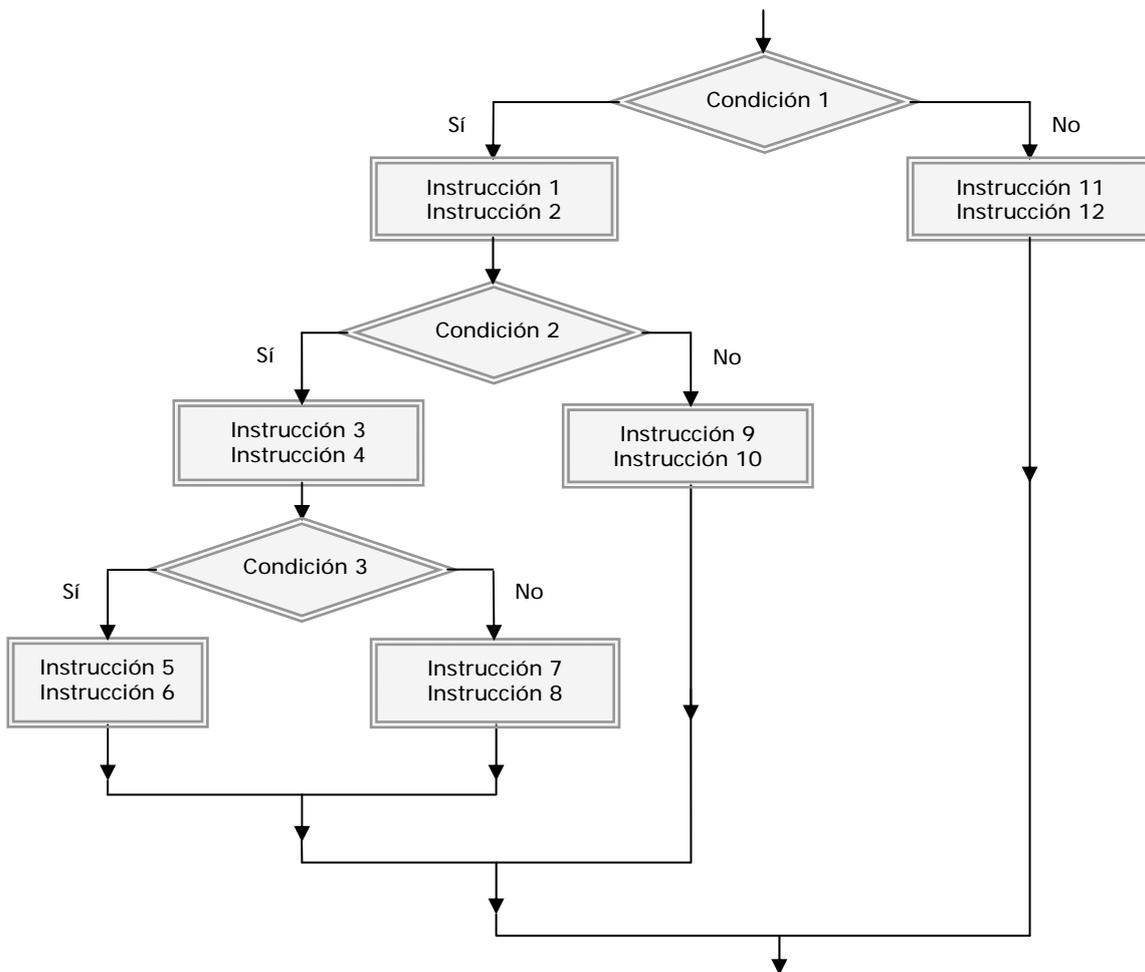
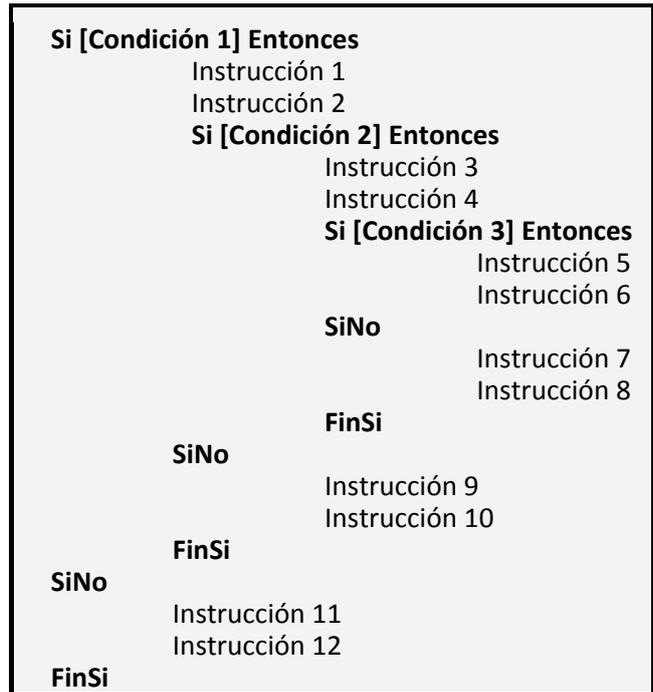
Válido

La instrucción *Si ... Entonces* es anidable dentro de sí misma. Esto significa que puede haber un bloque *Si ... Entonces* dentro de otro. Se hace más necesario que nunca el sangrado para que el pseudocódigo sea legible. Ha de verificarse que todo *Si* se cierra con un *FinSi*.

Anidamiento simple (pseudocódigo y diagrama de flujo)



Anidamiento doble (pseudocódigo y diagrama de flujo)



Los anidamientos se pueden convertir en triples, cuádruples, quíntuples, etc. lo cual convierte en farragosa la lectura y seguimiento del pseudocódigo. Habrá que tratar de evitar excesos usando otros recursos, desviando a módulos independientes, etc.

Cuando sea necesario por la complejidad en la toma de decisiones, se recurrirá a una tabla de decisión como paso previo a la confección del pseudocódigo o diagrama de flujo.

Próxima entrega: CU00143A

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