



aprenderaprogramar.com

# Pseudocódigo ejemplo de uso de bucles for-next y condicionales if-then en programación.

## (CU00164A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 63 del Curso Bases de la programación Nivel I

24

## EJERCICIO

En un complejo industrial se desea incentivar a los encargados de turnos de producción. A cada encargado le corresponde un turno. Hay 18 turnos, y por cada uno de ellos 40 operarios. Para evaluar a los encargados se ha hecho una encuesta entre los trabajadores. Se les ha pedido que evalúen a su encargado de 0 a 10 siendo el 5 el aprobado. Los resultados están almacenados en un array tipo *Nota(turno, operario)* donde turno indica el número de turno y operario el número de operario dentro del turno. Desarrollar el pseudocódigo y diagrama de flujo para un algoritmo que determine en qué turnos todos los operarios del mismo han aprobado al encargado.

## SOLUCIÓN

### Organización de variables:

Variable	Descripción - Uso
i, j	Uso como contadores
Nota(i,j)	Array de variables que almacenan la puntuación otorgada al encargado por el operario nº j del turno nº i
Haysuspensos	Interruptor que vale verdadero si algún operario del turno suspende al encargado y falso en caso contrario

### Pseudocódigo:

<p>1. Inicio [Evaluación de encargados aprenderaprogramar.com]</p> <p>2. Desde i = 1 hasta 18 Hacer</p> <p>    2.1 Desde j = 1 hasta 40 Hacer</p> <p>        Leer Nota(i,j)</p> <p>        Si Nota(i,j) &lt; 5 Entonces</p> <p>            Haysuspensos = Verdadero</p> <p>        FinSi</p> <p>    Siguiete</p> <p>    2.2 Si Haysuspensos = Falso Entonces</p> <p>        Mostrar "Todos los operarios del turno", i, "han aprobado al encargado"</p> <p>    FinSi</p> <p>    2.3 Haysuspensos = Falso</p> <p>    Siguiete</p> <p>3. Fin</p>
--

**Comentarios:** Nótese cómo el uso de un interruptor simplifica una evaluación de múltiples condicionantes.

La evaluación propuesta es equivalente a:

```
Si Nota(i,j) < 5 ó Nota(i,j+1) < 5 ó Nota(i,j+2) < 5 ... ó Nota(i,40) < 5 Entonces
    Haysuspensos = Verdadero
FinSi
```

Obviamente con los recursos que conocemos ya no tiene sentido pensar en este tipo de enumeración de condiciones.

Otro aspecto interesante es que cuando se han extraído las conclusiones derivadas del estado del interruptor y se va a empezar un nuevo ciclo, hay que “apagar” el interruptor. En caso contrario nos enfrentamos a conclusiones erróneas, de las que se pueden derivar los ya consabidos malfuncionamientos, errores, bloqueos, etc. Por tanto, precaución frente a este tipo de olvidos pues es también de los habituales. En un programa largo y complejo, puede hacerse difícil dar con el interruptor que está fallando.

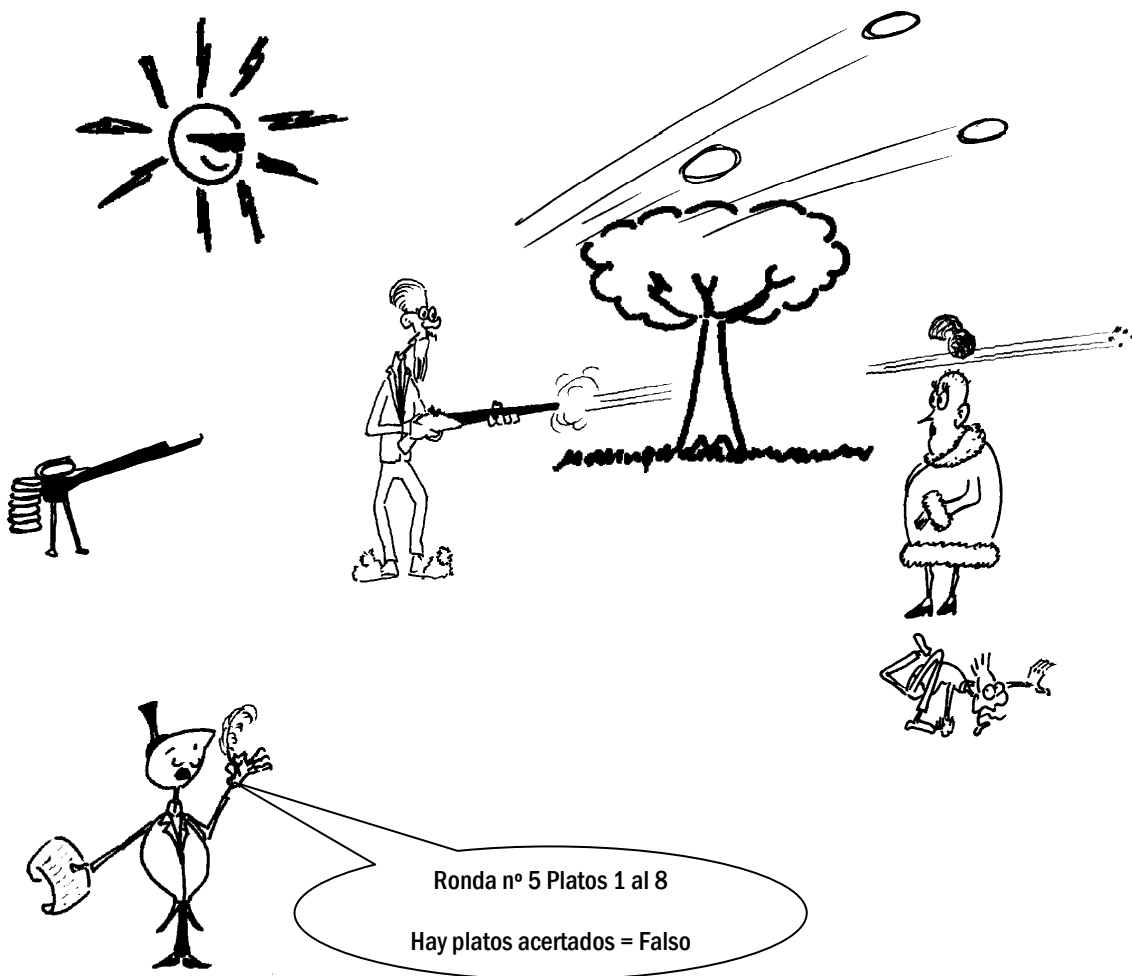
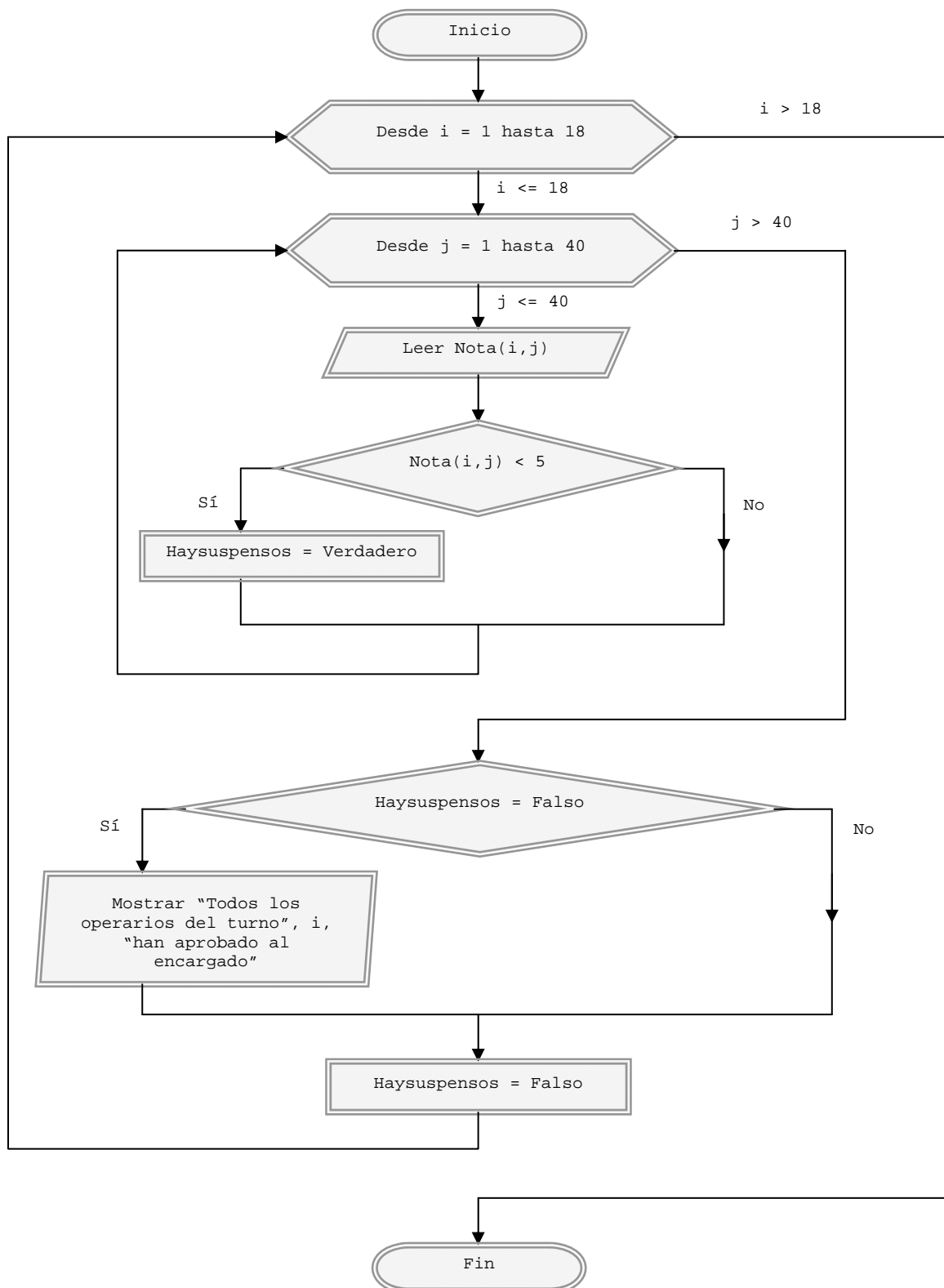


Ilustración: Utilizamos una instrucción Desde ... Siguiente cuando el número de iteraciones es definido. A su vez un interruptor nos sirve para obtener información de lo que ha ocurrido durante el proceso de datos.

**Diagrama de flujo:**



**Próxima entrega: CU00165A**

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