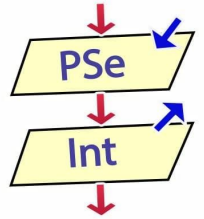
Agencia de Aprendizaje a lo largo de la vida

CODO A CODO INICIAL
Clase 6

Algoritmia 6

BA

Estructuras de control II
Condicionales anidados



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 05

Algoritmia 5 - Estructuras condicionales I

- Operadores relacionales
- Estructuras condicionales en programación.
- Si...Entonces y Si...Entonces...Sino

Clase 06

Algoritmia 6 - Estructuras condicionales II

- Estructuras no secuencuenciales en programación.
- Anidamiento de estructuras: uso de condicionales anidados.

Clase 07

Algoritmia 7 - Estructuras condicionales III

- Uso de la estructura Segun... en lugar de condicionales múltiples.

Anidamiento de estructuras

El anidamiento de condicionales en programación se refiere a la práctica de incorporar estructuras condicionales dentro de otras, formando una jerarquía de evaluaciones. Esta técnica permite manejar situaciones complejas al tomar decisiones basadas en múltiples condiciones.
Es una herramienta poderosa, pero su uso debe equilibrarse. Se recomienda limitar el anidamiento para mantener un código claro y fácilmente comprensible. En situaciones simples, estructuras condicionales independientes pueden ser más apropiadas.

Anidamiento de condicionales | Ventajas

En anidamiento de condicionales, bien utilizado, presenta varias ventajas:

- **Granularidad:** Permite una clasificación detallada de casos al evaluar condiciones específicas en diferentes niveles.
- **Flexibilidad:** Facilita la adaptación del flujo del programa a situaciones variadas, mejorando la capacidad de respuesta.
- **Claridad en Casos Específicos:** Permite tratar casos específicos de manera detallada, lo que puede ser esencial en algoritmos más complejos.
- **Abstracción Jerárquica:** Crea una estructura jerárquica que refleja la lógica del problema, facilitando la comprensión del código.

Anidamiento de condicionales | Desventajas

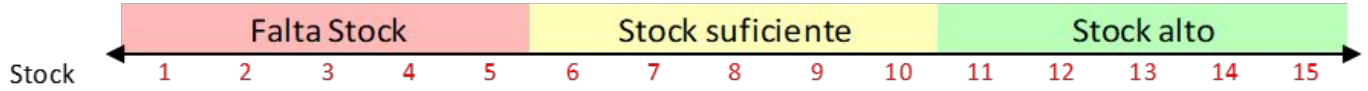
Sin embargo, su uso descuidado puede tener algunos inconvenientes:

- **Complejidad:** A medida que se anidan más condicionales, la complejidad del código aumenta, lo que puede dificultar la comprensión y mantenimiento.
- **Posible Confusión:** El anidamiento excesivo puede llevar a la confusión, especialmente si no se documenta adecuadamente.
- **Dificultad en la Depuración:** La identificación y corrección de errores puede volverse más compleja al usar anidamientos profundos.
- **Posibilidad de Redundancia:** Se corre el riesgo de redundancias y repeticiones en la lógica condicional.

Anidamiento de condicionales | Ejemplo I

Una mueblería necesita un programa que permita determinar el **nivel de stock*** de sus productos, de acuerdo a estas condiciones:

- Si la cantidad en stock es mayor a 10, se considera un “Stock alto”.
- Si la cantidad en stock es mayor o igual a 6, se considera un “Stock suficiente”.
- Si la cantidad en stock es menor a 6, se considera que “Falta Stock”.



** El "stock" se refiere a la cantidad de productos que la muebleria tiene disponibles para la venta en un momento dado.*

Anidamiento de condicionales | Ejemplo I

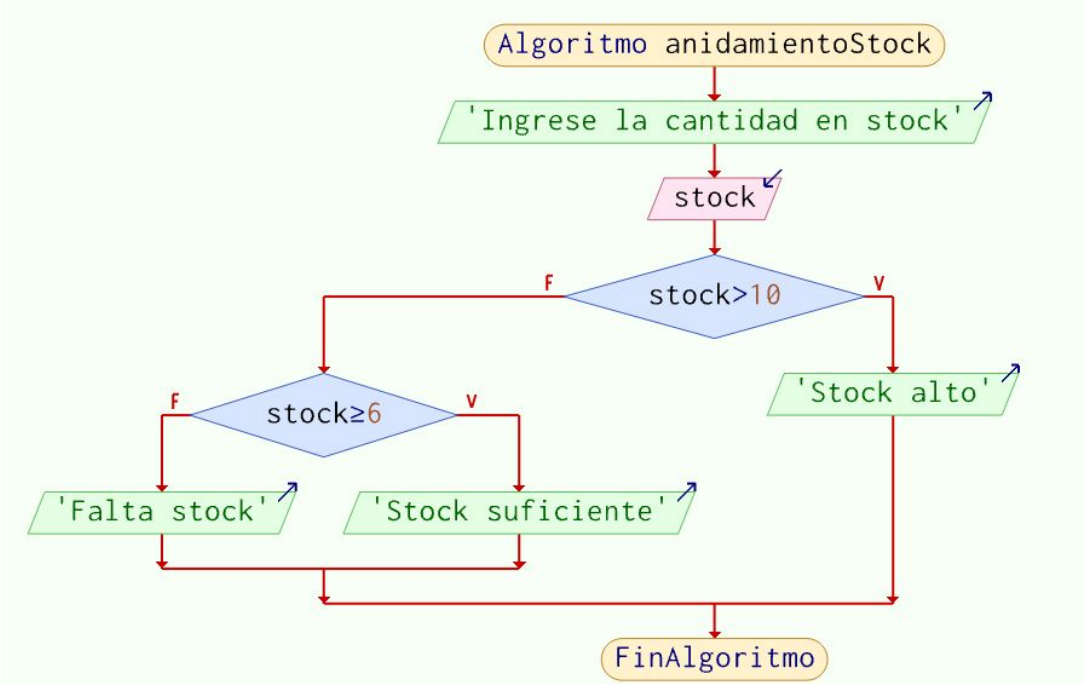
```
Algoritmo anidamientoStock

    Escribir "Ingrese la cantidad en stock"
    Leer stock

    Si stock > 10 Entonces
        Escribir "Stock alto"
    SiNo
        Si stock >= 6 Entonces
            Escribir "Stock suficiente"
        SiNo
            Escribir "Falta stock"
        FinSi
    FinSi

FinAlgoritmo
```

A

B

Este ejemplo utiliza condicionales anidados para comunicar mensajes específicos sobre el nivel de stock en función de las condiciones evaluadas.
El usuario ingresa la cantidad en stock y el programa evalúa varias condiciones utilizando condicionales anidados.
Primero, verifica si el **stock es mayor que 10 (A)**; si es así, imprime "*Stock alto*". En caso contrario, se “anida” otra estructura condicional, que evalúa si el **stock es mayor o igual a 6 (B)**; si lo es, imprime "*Stock suficiente*". Si no cumple con esta condición, el programa imprime "*Falta stock*".

Anidamiento de condicionales | Ejemplo I



Anidamiento de condicionales | Ejemplo II

Un colegio necesita un programa que permita determinar si sus estudiantes han aprobado una materia, de acuerdo a estas condiciones:

- Si la nota ingresada es 0 o es mayor a 10, se considera “Calificación no válida”.
- Si la nota ingresada es 10, se considera “Calificación Excelente”.
- Si la nota ingresada es mayor o igual a 6, se considera que el estudiante ha aprobado.
- En el resto de los casos, se considera que el estudiante ha reprobado.

El programa deberá imprimir por pantalla mensajes adecuados en cada caso.

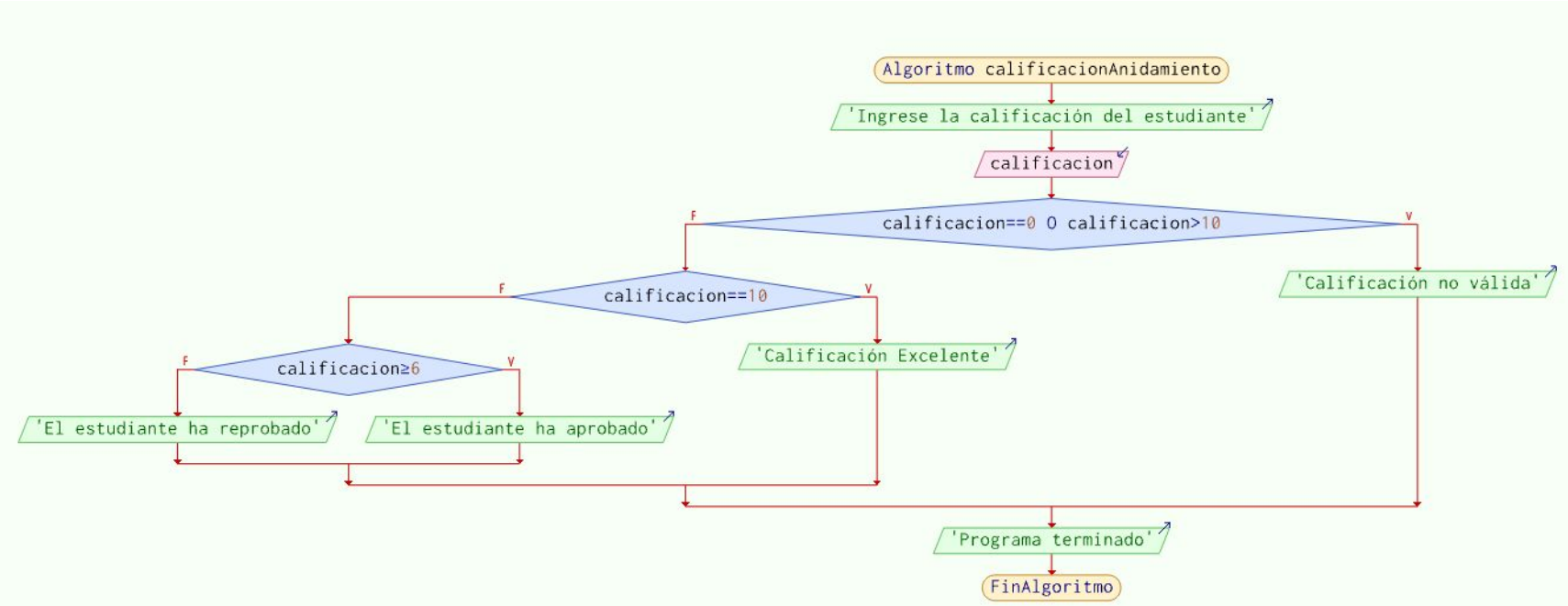
Anidamiento de condicionales | Ejemplo II

```
Algoritmo calificacionAnidamiento

    Escribir "Ingrese la calificación del estudiante"
    Leer calificacion
    Si calificacion == 0 O calificacion > 10 Entonces
        Escribir 'Calificación no válida'
    SiNo
        Si calificacion == 10 Entonces
            Escribir 'Calificación Excelente'
        SiNo
            Si calificacion >= 6 Entonces
                Escribir 'El estudiante ha aprobado'
            SiNo
                Escribir 'El estudiante ha reprobado'
            FinSi
        FinSi
    FinSi
    Escribir 'Programa terminado'

FinAlgoritmo
```

A
B
C

Luego de leer la calificación de un estudiante, se verifica si esta es igual a cero o mayor que diez **(A)**, en cuyo caso el programa imprime *'Calificación no válida'*. En caso contrario, se evalúa si la calificación es igual a diez **(B)**; si es así, se imprime '*Calificación Excelente*'. Si no es igual a diez, se verifica si la calificación es mayor o igual a seis **(C)**. Si se cumple esta condición, el programa imprime '*El estudiante ha aprobado*'. Si la calificación no cumple con ninguna de las condiciones anteriores, se imprime '*El estudiante ha reprobado*'. Finalmente, el programa muestra '*Programa terminado*', indicando el final de la ejecución.

Anidamiento de condicionales | Ejemplo II



Desafíos

Desafío 1: Clasificación de Triángulos

Escribe un programa que recibe tres longitudes como entrada y utiliza condicionales anidados para determinar el tipo de triángulo que se forma (equilátero, isósceles o escaleno).

Recuerda:

- Triángulo Equilátero: Todos sus lados son de igual longitud.
- Triángulo Isósceles: Posee al menos dos lados de igual longitud.
- Triángulo Escaleno: Todos sus lados tienen longitudes diferentes

Desafío 2: El problema del salario

Escribe un programa que solicite al usuario ingresar la calificación numérica de un estudiante. Utilizando condicionales anidados, asigna una calificación (A, B, C, D, F) basada en el siguiente sistema predefinido:

- ***A**: Calificaciones entre 90 y 100.*
- ***B**: Calificaciones entre 80 y 89.*
- ***C**: Calificaciones entre 70 y 79.*
- ***D**: Calificaciones entre 60 y 69.*
- ***F**: Calificaciones entre 0 y 59.*

El programa debe evaluar la calificación numérica ingresada y asignar la calificación correspondiente.

Desafío 3: Categorización de Edades

Crea un programa que solicite la edad de una persona como entrada y utilice condicionales anidados para categorizar la edad en grupos (niño, adolescente, adulto, adulto mayor).

Elije tu mismo los rangos correspondientes para cada categoría.

Desafío 4: Piedra, papel o tijeras

*Escribe un programa en PSeInt que simule el juego de **piedra, papel o tijeras** contra la computadora.*

El programa debe solicitar al usuario que elija entre piedra (1), papel (2) o tijeras (3). La computadora seleccionará aleatoriamente una de las opciones.

Luego, se determinará el ganador según las reglas del juego: piedra vence tijeras, tijeras vencen papel y papel vence piedra.

El programa debe mostrar el resultado del juego, indicando la elección del usuario, la elección de la computadora y si el usuario ganó, perdió o empató.

Desafío 5: Desigualdad triangular

Los tres lados a, b y c de un triángulo deben satisfacer la **desigualdad triangular**: cada uno de los lados no puede ser más largo que la suma de los otros dos.

Sabiendo esto, desarrolle un programa que, al recibir como entrada las longitudes de los tres lados de un triángulo, evalúe si cumple con la desigualdad triangular. En caso de que el triángulo sea inválido, el programa deberá indicarlo; de lo contrario, determinará si el triángulo es equilátero, isósceles o escaleno.

Material extra

Artículos de interés

Videos:

- Condicionales anidadas en Pseint (Sino Si) | Coders Free
- PSeint - Condicionales Anidadas - Ejercicio Cálculo IMC | DTB programación

No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.
- Realizar el Ejercicio de Repaso.

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto.