

ESTRUCTURAS DE CONTROL CON PSEINT – ESTRUCTURAS SELECTIVAS





Objetivos de la Guía

En esta guía aprenderemos a:

- Diferenciar estructuras secuenciales de selectivas.
- Armar estructuras condicionales.
- Usar estructuras condicionales.

GUÍA DE ESTRUCTURAS DE CONTROL

En nuestra primera guía de PSeInt nuestros algoritmos consistieron en simples secuencias de instrucciones unas después de otra. Estos algoritmos, con una instrucción detrás de otra se conocen como estructuras secuenciales, pero en nuestros programas existen tareas más complejas que no pueden ser resueltas así, quizás necesitamos repetir una misma instrucción, realizar acciones diferentes en función del valor de una expresión, etc. Para esto existen las estructuras de control.

¿QUÉ ES UNA ESTRUCTURA DE CONTROL?

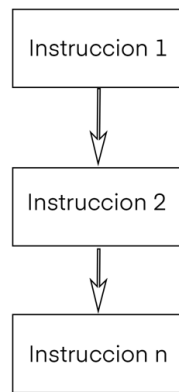
Las **Estructuras de Control** determinan el orden en que deben ejecutarse las instrucciones de un algoritmo, es decir, **si serán recorridas una después de la otra** (estructuras secuenciales), **si habrá que tomar decisiones sobre si ejecutar o no alguna acción** (estructuras selectivas o de decisión) o **si habrá que realizar repeticiones** (estructuras repetitivas). Esto significa que una estructura de control permite que se realicen unas instrucciones y omitir otras, de acuerdo a la evaluación de una condición.

Esto hace que las estructuras de control se puedan dividir en tres:

- Estructuras secuenciales.
- Estructuras selectivas o de decisión.
- Estructuras repetitivas.

¿QUÉ ES UNA ESTRUCTURA SECUENCIAL?

Es la estructura en donde una acción (instrucción) **sigue a otra de manera secuencial**. Las tareas se dan de tal forma **que la salida de una es la entrada de la que sigue** y así en lo sucesivo hasta cumplir con todo el proceso. Esta estructura de control es la más simple, permite que las instrucciones que la constituyen se ejecuten una tras otra en el orden en que se listan.

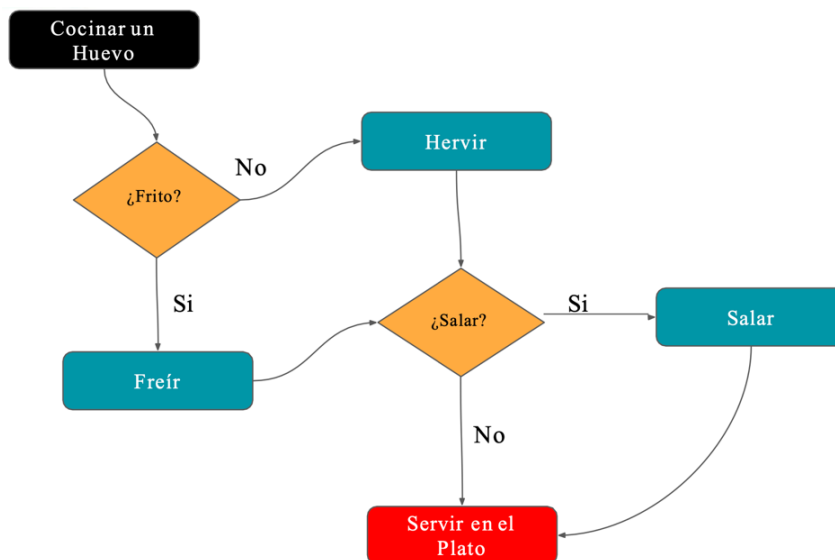


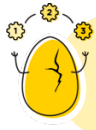
¿QUÉ ES UNA ESTRUCTURA SELECTIVA?

Estas estructuras de control son de gran utilidad para cuando el algoritmo a desarrollar requiera una descripción más complicada que una lista sencilla de instrucciones. Este es el caso cuando existe un número de posibles alternativas que resultan de la evaluación de una determinada condición. Este tipo de estructuras son utilizadas para tomar decisiones lógicas, es por esto que también se denominan **estructuras de decisión o selectivas**.

En estas estructuras, se realiza una **evaluación de una condición y de acuerdo al resultado, el algoritmo realiza una determinada acción**. Las condiciones son especificadas utilizando expresiones lógicas.

Una estructura selectiva, con varias condiciones, sería el ejemplo que usamos en la primera guía de **Cocinar un Huevo**. Las preguntas serían las condiciones a evaluar y de acuerdo a ese resultado realiza una o u otra acción.





El tiempo compartido vale

No tengas miedo en preguntar a quienes van más adelantados. Ellos al explicar reafirman ideas o se dan cuenta que no dominan el concepto. En un futuro, serás tú quien destinará minutos de su tiempo a explicar a un compañero.

¿QUÉ ES UNA CONDICIÓN?

En programación, una condición es toda **sentencia** de la cual se puede determinar **su verdad** (true) o **falsedad** (false). En su gran mayoría, son comparaciones. Por ejemplo, $4 > 5$, ésta sentencia **es una condición porque tiene resultado verdadero o falso**, en este caso falso porque 4 no es mayor a 5. En cambio, la siguiente sentencia, **Escribir "EggEducacion"**, **no es condición** puesto que no hay para comparar, no se puede determinar verdad o falsedad.

Podemos usar como condición el valor de una variable lógica que tengamos definida, ya que es lo mismo: verdadero o falso.

Por lo que una condición sirve para discernir entre una opción u otra, y en el proceso mental normalmente se manifiesta con un "Si"; por ejemplo: Si (va a llover), coge el paraguas.

Para determinar condiciones, precisamos utilizar Operadores.

¿Qué son los operadores?

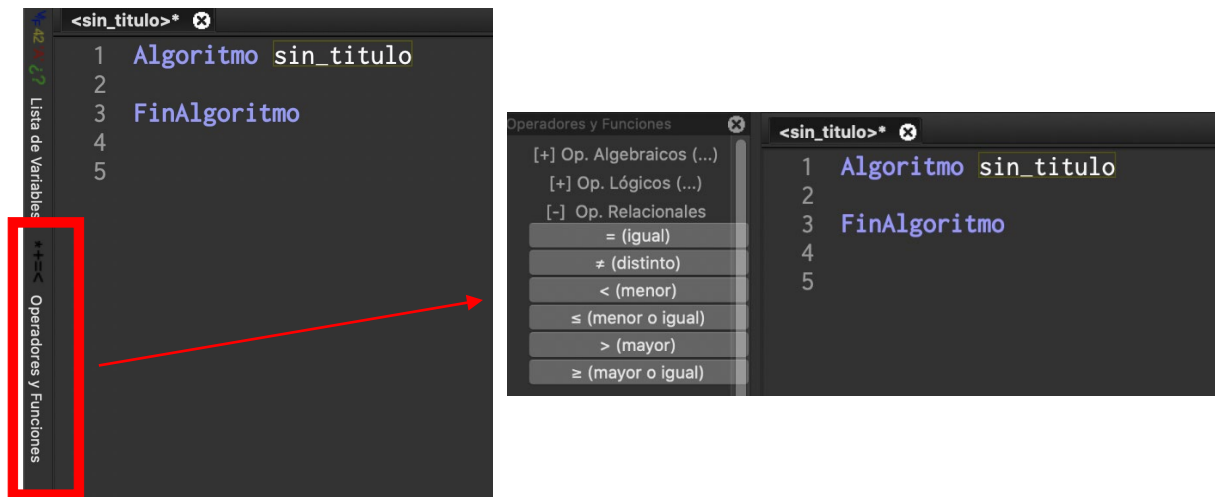
Las condiciones que usaremos en las estructuras selectivas y el resto de nuestras estructuras de control se realizan con la ayuda de los operadores relacionales y lógicos.

OPERADORES RELACIONALES

Los operadores relacionales son símbolos que se usan para comparar dos valores. Si el resultado de la comparación es correcto la expresión considerada es verdadera, en caso contrario es falsa.

Operador	Significado	Ejemplo
Relacionales		
>	Mayor que	$3 > 2$ // verdadero
<	Menor que	$1 < 5$ // verdadero
==	Igual que	$4 == 4$ // verdadero
>=	Mayor o igual que	$4 >= 5$ // falso
<=	Menor o igual que	'a' <= 'b' // verdadero
<>	Distinto que	$10 <> 8$ // verdadero

Estos operadores los podemos encontrar en el menú izquierdo de PseInt o podemos cargarlos por teclado.



OPERADORES LÓGICOS

Estos se utilizan cuando necesitamos las expresiones lógicas con múltiples variantes y nos proporcionan un resultado a partir de que se cumpla o no una cierta condición, estos producen un resultado lógico, y sus operadores son también valores lógicos o asimilables a ellos.

Operador	Significado	Ejemplo
Lógicos		
Y	Conjunción	(2 < 4 Y 3 > 5) // falso
O	Disyunción	(7 <= 8 O 10 >= 9) // verdadero
NO / no	Negación	no(1 == 1) // falso

Operador Y

Devuelve un valor lógico verdadero si ambas expresiones son verdaderas. En caso contrario el resultado es falso.

Operador O

Este operador devuelve verdadero si alguna de las expresiones es verdadera. En caso contrario devuelve "falso".

Operador NO

Este operador cambia la devolución de una expresión, al caso contrario. Si es verdadero lo hace falso y si es falso lo hace verdadero.

A la hora de trabajar con operadores lógicos, para saber si una expresión lógica nos devuelve como resultado Verdadero o Falso, debemos observar la siguiente **tabla de la verdad**:

Conjunción

A	Operador	B	Resultado
V	Y	V	V
V	Y	F	F
F	Y	V	F
F	Y	F	F

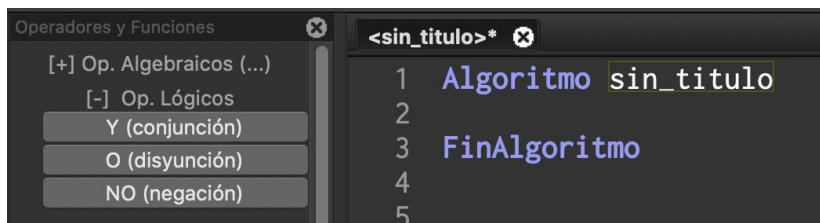
Disyunción

A	Operador	B	Resultado
V	O	V	V
V	O	F	V
F	O	V	V
F	O	F	F

Negación

A	Resultado	B	Resultado
no(V)	F	no(F)	V

Estos operadores también los podemos encontrar en el menú izquierdo de PSeInt o podemos cargarlos por teclado.



Reglas de prioridad:

Las expresiones que tienen dos o más operadores requieren unas reglas matemáticas que permitan determinar el orden de las operaciones, se denominan reglas de prioridad y son:

1. Las operaciones que están encerradas entre paréntesis se evalúan primero. Si existen diferentes paréntesis anidados (interiores unos a otros), las expresiones más internas se evalúan primero.
2. Las operaciones lógicas dentro de una expresión suelen seguir el siguiente orden de prioridad:
 - ❑ operador ()
 - ❑ operador negación NO
 - ❑ operador conjunción Y
 - ❑ operador disyunción O

En caso de coincidir varios operadores de igual prioridad en una expresión o sub expresión encerrada entre paréntesis, el orden de prioridad en este caso es de **izquierda a derecha**, y a esta propiedad se denomina asociatividad.



Vamos a mostrar ejemplos de condiciones tanto con operadores relacionales, como con lógicos

$x=y$, significa "si x es igual a y "

$x>y$, significa "si x es mayor que y "

$x<y$, significa "si x es menor que y "

$x\neq y$, significa "si x es distinto de y "

$(x==j) \text{ Y } (x==z)$, significa "si x es igual a j Y x igual a z "

$(x==y) \text{ O } (x==z)$, significa "si x es igual a j O x igual a z "



EJERCICIO OPERADORES

Vamos a poner en práctica los usos de los operadores, para ello definiremos una variable de tipo lógico y utilizaremos los operadores lógicos y relacionales para otorgarle un valor y mostrarlo por pantalla.

```
1  Algoritmo PRACTICA_OPERADORES
2
3      Definir bandera Como Logico
4
5      bandera = 4 > 5
6
7      Escribir bandera
8
9  FinAlgoritmo
10
11
```

PSelnt - Ejecutando proceso PRACTIC

```
*** Ejecución Iniciada. ***
FALSO
*** Ejecución Finalizada. ***
```

Prueba tantos operadores lógicos como relacionales quieras. Intenta aplicar las reglas de prioridad y validar varias condiciones a la vez con operadores lógicos.

DETECCIÓN DE ERRORES

Copia y pega este código en tu programa. Deberás corregir los errores hasta lograr el siguiente resultado esperado:

```
Algoritmo PRÁCTICA_OPERADORES
bandera = 4 < 5
Escribir bandera
```

```
bandera = (2 12 MOD 2) 0 (NO 3 5 0 32 3^5)
```

```
Escribir bandera
```

```
bandera = 3 > 2 5 >= 3
```

```
FinAlgoritmo
```

¿Cuál es el resultado a lograr?

```
*** Ejecución Iniciada. ***  
FALSO  
VERDADERO  
VERDADERO  
*** Ejecución Finalizada. ***
```



Revisemos lo aprendido hasta aquí

- Diferenciar estructuras secuenciales de las selectivas.
- Elaboración de condiciones
- Implementación de operadores relacionales y lógicos para la creación de condiciones.

¿CUÁLES SON LAS ESTRUCTURAS SELECTIVAS?

Entendemos que las estructuras selectivas son utilizadas para tomar decisiones lógicas, es por esto que también se denominan **estructuras de decisión o selectivas**. Pero, ¿cuáles son las estructuras selectivas?

Las estructuras selectivas/alternativas pueden ser:

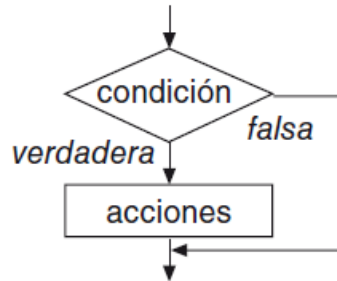
- Simples: Si
- Doble: Si- SiNo
- Múltiples: Según – Si Anidado



Para evitar errores de sintaxis, recomendamos seleccionar las estructuras selectivas desde el panel de comandos de la derecha en PSeInt.

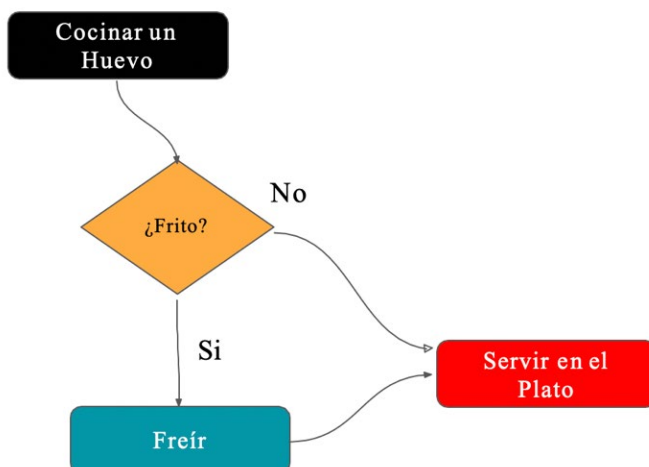
CONDICIÓN SIMPLE

La estructura alternativa simple si-entonces lleva a cabo una acción siempre y cuando se cumpla una determinada condición.



La selección si-entonces evalúa la condición y luego:

- Si la condición es verdadera, ejecuta el bloque de acciones
- Si la condición es falsa, no ejecuta otra opción.



Si fuera cocinar un huevo, tenemos solo la opción de freírlo y si no lo queremos frito, se va servir crudo en el plato. Esto es una **Condición Simple**.

Condición Simple en PSeInt:

```
Si condicion Entonces
    acciones_por_verdadero
Fin Si
```



Pueden encontrar un ejemplo para descargar de Condición Simple en el Aula Virtual.



MANOS A LA OBRA!

EJERCICIO FACILITADOR

Escriba un programa en donde se le pregunte al estudiante si el día de hoy le toca ser el facilitador de tu equipo. En caso de que sea, que muestre por pantalla el siguiente mensaje: "¡Felicidades! Eres el facilitador de tu equipo."

DETECCIÓN DE ERRORES

Copia y pega este código en tu programa. Deberás corregir los errores hasta lograr el siguiente resultado esperado:

Algoritmo PRACTICA_CONDICIONAL_SI

bandera = 2 <= 1

Si band Entonces

 Escribir "La condición es: " " por lo tanto la ejecución entra en la estructura selectiva Si y ejecuta esta línea "

Fin Si

FinAlgoritmo

¿Cuál es el resultado a lograr?

No podemos escribir Verdadero en el escribir, deberemos llegar a ese resultado a través del código.

```
PSelnt - Ejecutando proceso PRACTICA_CONDICIONAL_SI
*** Ejecución Iniciada. ***
La condición es: VERDADERO por lo tanto la ejecución entra
en la estructura selectiva Si y ejecuta esta línea
*** Ejecución Finalizada. ***
```



Revisemos lo aprendido hasta aquí

- Crear la Estructura Sí, permitiendo al programa ejecutar un bloque de código si la condición que estableciste es verdadera.

Si no pudiste interiorizar el concepto en su totalidad, **no te preocupes**, más adelante seguiremos trabajando sobre este tema.