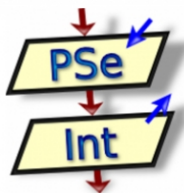


PROGRAMACIÓN DESDE CERO

MATERIAL DE TRABAJO

# Estructuras de control con PseInt – Estructuras selectivas

ENCUENTRO 5





## 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

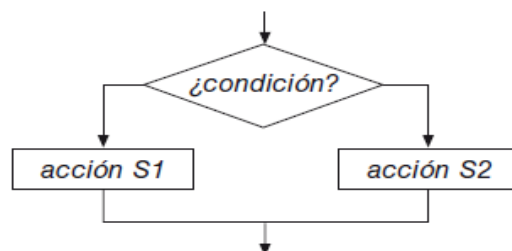
En la guía anterior vimos que las estructuras selectivas son utilizadas para tomar decisiones lógicas, es por esto que también se denominan **estructuras de decisión o selectivas**. Y que existen tres tipos:

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

La última vez **solo vimos las simples**, por lo que hoy veremos una de las dos restantes, las dobles.

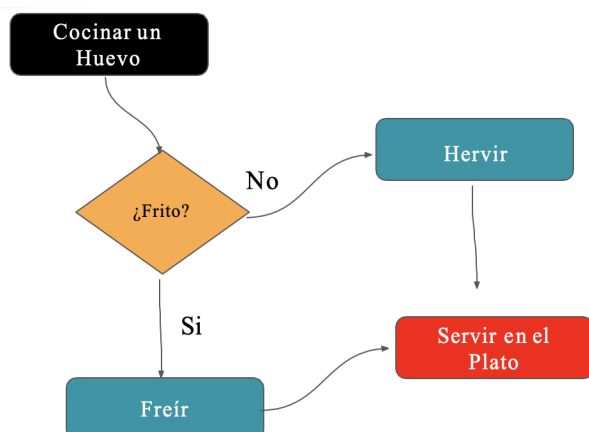
### CONDICIÓN DOBLE

La estructura anterior es muy limitada y normalmente se necesitará una estructura que permita elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición. Si la condición es verdadera, se ejecuta la acción S1 y, si es falsa, se ejecuta la acción S2.



La selección si-entonces-sino evalúa la condición y luego:

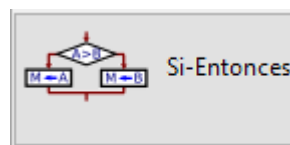
- Si la condición es verdadera, ejecuta el bloque de acciones
- Si la condición es falsa, ejecuta el bloque de acciones 2.



En este caso, si fuera cocinar un huevo, tenemos opción de freírlo y si no lo queremos frito, tendremos la opción de hervirlo. Esto es una **Condición Doble**.

#### Condición Doble en PSeInt

```
Si condicion Entonces
..... acciones_por_verdadero
SiNo
..... acciones_por_falso
Fin Si
```



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



#### EJERCICIO ZOOM

Realizar un programa que pida al usuario el horario en el que se conectó al zoom. Si ese horario está entre la hora de ingreso y la hora de ingreso + los 15' de tardanza, mostrará un mensaje por pantalla que diga "Llegaste a tiempo, tienes presente". Si el horario de ingreso supera ese límite, se mostrará un mensaje por pantalla que diga "Hoy tendrás tardanza. Recuerda avisarle a tus coaches cuando llegarás tarde. Mañana te esperamos a tiempo, tu participación en el equipo es VITAL"

## DETECCIÓN DE ERRORES

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

Algoritmo PRÁCTICA CONDICIONAL SINO

Definir bandera Como Logico

bandera = 2 > 3   3 == 3

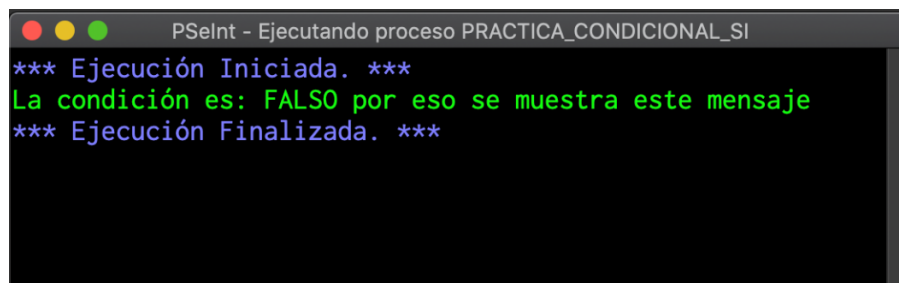
Si bandera Entonces

    Escribir "La condición es: " bandera " por eso se muestra este mensaje"

SiNo

    Escribir "La condición es: " bandera " por eso se muestra este mensaje"

¿Cuál es el resultado a lograr?



```
*** Ejecución Iniciada. ***
La condición es: FALSO por eso se muestra este mensaje
*** Ejecución Finalizada. ***
```



### Revisemos lo aprendido hasta aquí

- Implementar la Estructura Si-Sino, permitiendo al programa ejecutar un bloque de código si la condición que estableciste es verdadera y otro bloque distinto si la condición es falsa

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

## FUNCIONES PSEINT

Además de empezar a implementar las estructuras de control, vamos a empezar a utilizar las **funciones de PSeInt**. Las funciones, son herramientas que nos proporciona PSeInt y cumplen el propósito de *ayudarnos a resolver ciertos problemas*. Supongamos que tenemos que calcular la raíz cuadrada de un número, PSeInt cuenta con una función que, pasándole un número, nos devuelve el resultado de su raíz cuadrada. Ese resultado que devuelve, se lo podemos asignar a una variable o lo podemos concatenar con un escribir para mostrar el resultado sin la necesidad de una variable.

También, las *funciones se pueden utilizar dentro de cualquier expresión, de cualquier estructura, y cuando se evalúe la misma, se reemplazará por el resultado correspondiente*.

Tenemos dos tipos de funciones, funciones matemáticas y funciones de cadenas de texto. Las funciones matemáticas, reciben un sólo parámetro de tipo numérico y devolverán un solo valor de tipo numérico. Las funciones de cadenas, en cambio, reciben un solo parámetro de tipo cadena, pero pueden devolver un valor de tipo cadena o de tipo numérico según la función que se use.

Funciones	Significado
RC(número)	Devuelve la raíz cuadrada del número.
ABS(número)	Devuelve el valor absoluto del número
LN(número)	Devuelve el logaritmo natural del número
EXP(número)	Devuelve la función exponencial del número.
SEN(número)	Devuelve el seno de número.
COS(número)	Devuelve el coseno de número.
TAN(número)	Devuelve la tangente de número.
ASEN(número)	Devuelve el arcoseno de número.
ACOS(número)	Arcocoseno de x
ATAN(número)	Arcotangente de x
MOD	Devuelve el módulo (resto de la división entera).
TRUNC(número)	Trunca el valor x (parte entera de x)
REDOND(número)	Redondea al valor más cercano a x
AZAR(número)	Entero aleatorio entre 0 y x -1
ALEATORIO(min,max)	Entero aleatorio entre valor mínimo y máximo



¿NECESITAS UN EJEMPLO?

Escribir "Raíz cuadrada de 9: " `rc(9)`

Escribir "Resto de 4/2: " `4 MOD 2`

Escribir "Valor absoluto de -3: " `abs(-3)`

Escribir "Seno de 90 grados: " `sen(90 * PI / 180)`

Escribir "Truncamos 3.7: " `trunc(3.7)`

Escribir "Redondeamos 2.7: " `redon(2.7)`

Escribir "Un número al azar del 0 al 9: " `azar(10)`

Escribir "Un número al azar entre 10 y 20: " `aleatorio(10,20)`

**Del código anterior los resultados serían:**

Raíz cuadrada de 9: `3`

Resto e 4/2: `0`

Valor absoluto de -3: `3`

Seno de 90 grados: `1`

Truncamos 3.7: `3`

Redondeamos 2.7: `3`

Un número al azar del 0 al 9: `6`

Un número al azar entre 10 y 20: `14`



Pueden encontrar un ejemplo para descargar de Funciones Matemáticas en el Aula Virtual.

## FUNCIONES CADENAS DE TEXTO

Algunas funciones de cadenas de texto utilizan las posiciones de cada letra de una cadena. Esto significa que, si tengo la palabra Hola, la cadena tendrá 4 posiciones, en PSeInt las posiciones de las letras arrancan en 0. Entonces para la cadena Hola, nuestras posiciones serían: 0: **H**, 1: **o**, 2: **l** y 3: **a**.

Funciones	Significado
Longitud(cadena)	Devuelve la cantidad de letras que compone la cadena.
Mayusculas(cadena)	Devuelve una copia de la cadena con todas sus letras en mayúsculas.

Minusculas(cadena)	Devuelve una copia de la cadena con todas sus letras en minúsculas.
Subcadena(cadena, posición_inicial, posición_final)	Devuelve una nueva cadena que consiste en la parte de la cadena que va desde la posición pos_inicial hasta la posición pos_final.
Concatenar(cadena, cadena2)	Devuelve una nueva cadena que resulta de unir las cadenas cadena1 y cadena2.
ConvertirANumero(cadena)	Recibe una cadena compuesta de números y devuelve la cadena como una variable numérica.
ConvertirACadena(cadena)	Recibe un número y devuelve una variable cadena de caracteres de dicho número.



Definir cadena1, cadena2 como cadena

cadena1 = "programacion"

cadena2 = "EGG"

Escribir "La longitud de cadena1 es: " longitud(cadena1)

Escribir "El primer carácter de cadena1 es: " subcadena(cadena1,0,0)

Escribir "La cadena1 en mayúsculas es: " mayusculas(cadena1)

Escribir "La cadena2 en minusculas es: " minusculas(cadena2)

Escribir "La cadena concatenada queda como: " concatenar(cadena1," es muy interesante")

Escribir "La cadena convertida a numero queda:" convertirANumero("10")

**Del código anterior los resultados serían:**

La longitud de cadena1 es: 12

El primer carácter de cadena1 es: p

La cadena1 en mayúsculas es: PROGRAMACION

La cadena2 en minúsculas es: egg

La cadena concatenada queda como: programacion es muy interesante

La cadena convertida a numero queda: 10



Pueden encontrar un ejemplo para descargar de Funciones de Cadenas de Texto en el Aula Virtual.