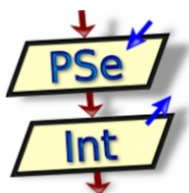


PROGRAMACIÓN DESDE CERO

ESTRUCTURAS DE CONTROL CON PSEINT - SELECTIVAS





Objetivos de la Guía

En esta guía aprenderemos a:

- Diferenciar estructuras secuenciales de selectivas.
- Armar estructuras condicionales.
- Usar estructuras condicionales.
- Desarrollar la cooperación trabajando en equipo.

GUÍA DE ESTRUCTURAS DE CONTROL – PARTE 1

En nuestra primera guía de Pselnt 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?

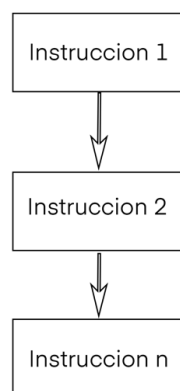
Las **Estructuras de Control** determinan el orden en que deben ejecutarse las instrucciones de un algoritmo, es decir, **si serán recorridas una después de la otra** (estructuras secuenciales), **si habrá que tomar decisiones sobre si ejecutar o no alguna acción** (estructuras selectivas o de decisión) 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.

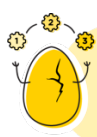
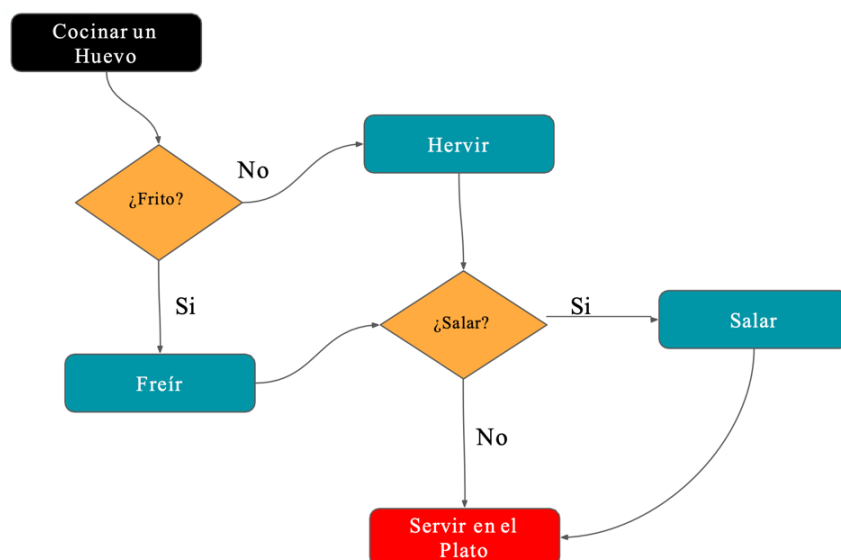


¿QUÉ ES UNA ESTRUCTURA SELECTIVA?

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 la gente que va más adelantada, esto no es una carrera, ¡nadie se atrasa! De hecho, les estás haciendo un favor, ya que 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.

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?

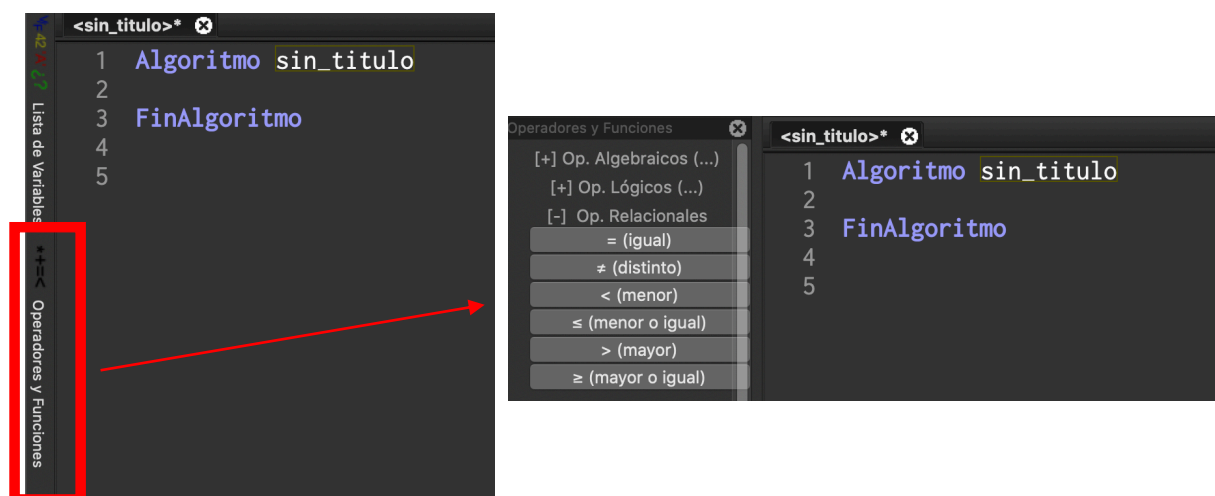
Las condiciones que usaremos en las estructuras selectivas y el resto de nuestras estructuras de control se realizan con la ayuda de los operadores relacionales y lógicos.

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	$(2 < 4 \text{ Y } 3 > 5)$ // falso
O	Disyunción	$(7 \leq 8 \text{ O } 10 \geq 9)$ // verdadero
NO / no	Negación	$\text{no}(1 == 1)$ // 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
$\text{no}(V)$	F	$\text{no}(F)$	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 ”

$x!=y$, 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 ”



MANOS A LA OBRA!

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:

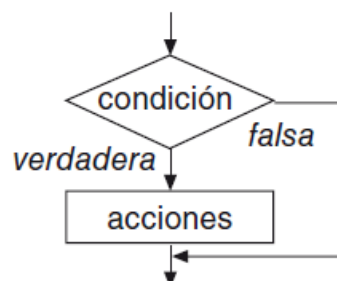
- Simples: Si
- Doble: Si- SiNo
- Múltiples: Según – Si Anidado



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.

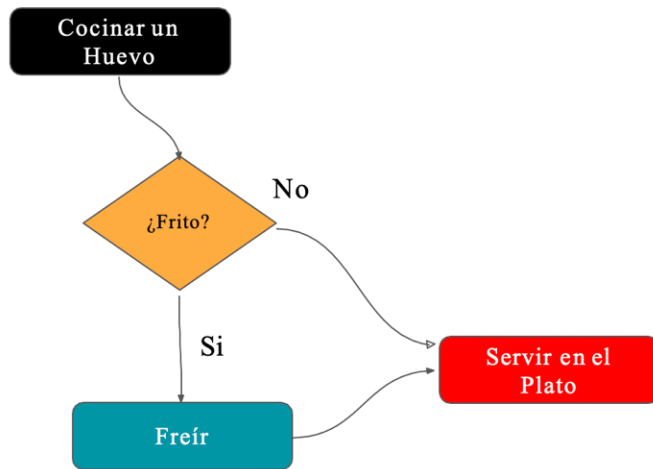


La selección si-entonces evalúa la condición y luego:

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



¿NECESITAS UN EJEMPLO?



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 Aula Virtual.



MANOS A LA OBRA!

EJERCICIO FACILITADOR

Escriba un programa en donde se le pregunte al alumno 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
```

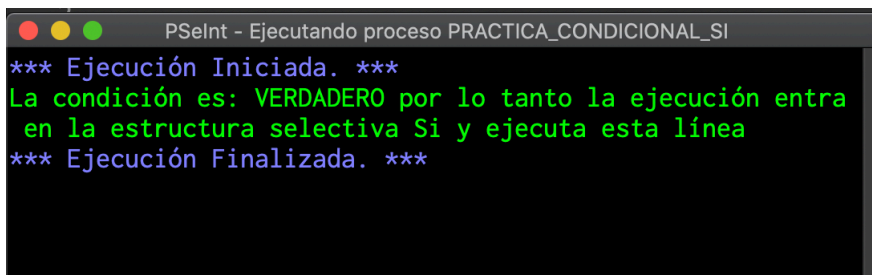
```
    Escribir "La condición es: " " por lo tanto la ejecución entra en la  
estructura selectiva Si y ejecuta esta línea "
```

```
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.



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



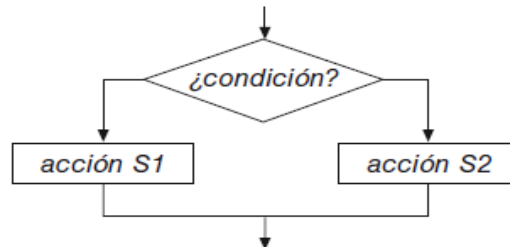
Revisemos lo aprendido hasta aquí

- 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.

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.

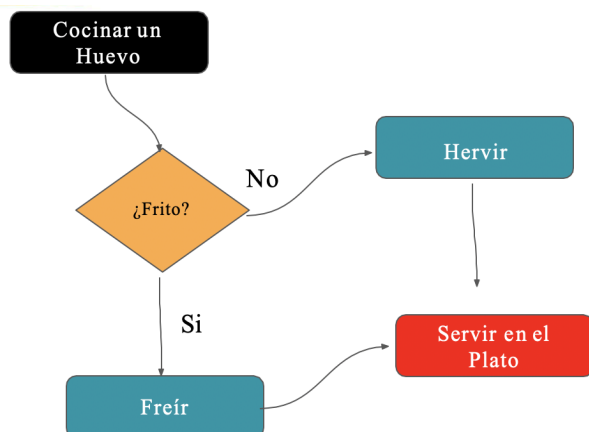


La selección si-entonces-sino evalúa la condición y luego:

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



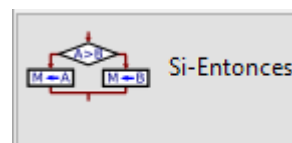
¿NECESITAS UN EJEMPLO?



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 Aula Virtual.



MANOS A LA OBRA!

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?

```
PSInt - Ejecutando proceso PRACTICA_CONDICIONAL_SI
*** Ejecución Iniciada. ***
La condición es: FALSO por eso se muestra este mensaje
*** Ejecución Finalizada. ***
```



Revisemos lo aprendido hasta aquí

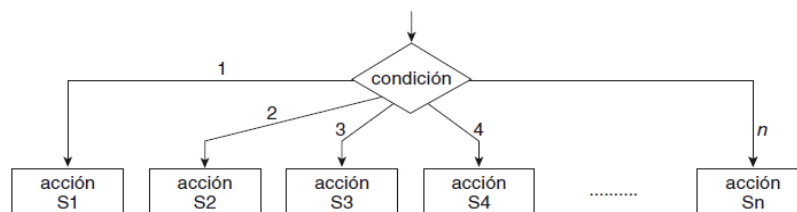
- Implementar la Estructura Si-Sino, permitiendo al programa ejecutar un bloque de código si la condición que estableciste es verdadera y otro bloque distinto si la condición es falsa

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

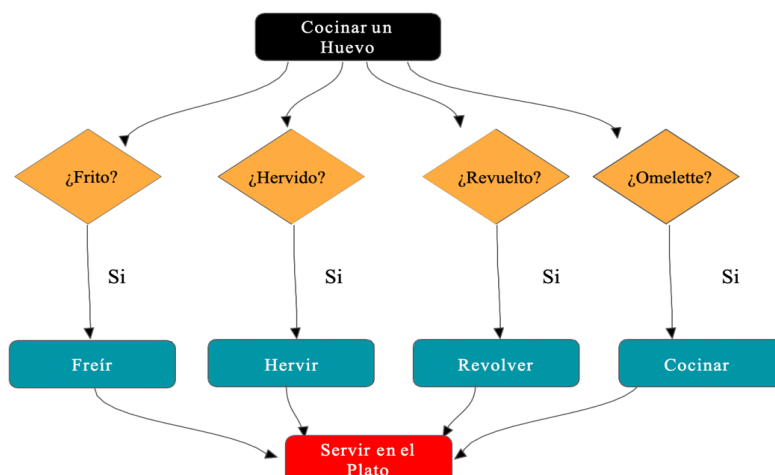
CONDICIÓN MÚLTIPLE

Muchas veces vamos a tener más de dos alternativas para elegir, o una variable que puede tomar varios valores. Para solucionar esto, usamos la condición múltiple. En esta estructura, se evalúa una condición o expresión que puede tomar n valores. Según el valor que la expresión tenga en cada momento se ejecutan las acciones correspondientes al valor.

La estructura de decisión múltiple evaluará una expresión que podrá tomar n valores distintos, 1, 2, 3, 4, ..., n . Según el valor que elija en la condición, se realizará una de las n acciones, o lo que es igual, el flujo del algoritmo seguirá un determinado camino entre los n posibles. Por ejemplo, si tenemos un sistema de notas, donde 6 es desaprobado, 7 es aprobado, 9 es sobresaliente y 10 es excelente. Al tener un valor que puede dar distintas alternativas, usamos la condición múltiple.



¿NECESITAS UN EJEMPLO?



Condición Multiple en PSeInt:



```
Segun variable_de_cualquier_tipo_de_dato Hacer
opcion_1:
    secuencia_de_acciones_1
opcion_2:
    secuencia_de_acciones_2
opcion_3:
    secuencia_de_acciones_3
De Otro Modo:
    secuencia_de_acciones_dom
Fin Segun
```

Este problema, se podría resolver por estructuras alternativas simples o dobles, anidadas o en cascada; sin embargo, este método si el número de alternativas es grande puede plantear serios problemas de escritura del algoritmo y naturalmente de legibilidad.



Cuando el valor de la variable que se **evalúa no coincide con ninguno de los valores que se evalúa**, entonces se ejecutan las acciones dentro del bloque “De Otro Modo” (secuencia_de_acciones_dom), el cual equivale a realizar un “Sino” dentro de las estructuras condicionales.

Además, pueden encontrar un ejemplo para descargar de Condición Multiple en Aula Virtual.



¿NECESITAS UN **EJEMPLO?**

```
1  Algoritmo PRACTICA_CONDICIONAL_SEGUN
2  Definir eleccion Como Caracter
3  Escribir "Ingrese una opción para cocinar su huevo"
4  Escribir "A- FRITO"
5  Escribir "B- HERVIDO"
6  Escribir "C- REVUELTO"
7  Escribir "D- OMELETTE"
8  Leer eleccion
9  Segun eleccion Hacer
10     "A":
11         Escribir "Su huevo se servirá FRITO"
12     "B":
13         Escribir "Su huevo se servirá HERVIDO"
14     "C":
15         Escribir "Su huevo se servirá REVUELTO"
16     "D":
17         Escribir "Su huevo se servirá OMELETTE"
18     De Otro Modo:
19         Escribir "La opción ingresada no está entre las ofrecidas"
20 Fin Segun
21 FinAlgoritmo
22
```

PSeInt - Ejecutando proceso PRACTICA_CONDICIONAL_SEGUN

```
*** Ejecución Iniciada. ***
Ingrese una opción para cocinar su huevo
A- FRITO
B- HERVIDO
C- REVUELTO
D- OMELETTE
> A
Su huevo se servirá FRITO
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible

Como podemos ver en el ejemplo, le damos al usuario varias opciones para elegir. El carácter que ingrese será analizado en la línea 9, y lo va comparando con las opciones disponibles. Si encuentra alguna coincidencia ejecutará las líneas de código dentro de esa opción, caso contrario se ejecutará el "De otro modo"



MANOS A LA OBRA!

EJERCICIO MESES

Ingresar un número del 1 – 12 y mostrar el mes del año que corresponde, si el número ingresado no es correcto mostrar un "mensaje de error".

DETECCIÓN DE ERRORES

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

Algoritmo PRACTICA_SEGUN

Definir num Como Caracter

Escribir "Ingrese un número entre 1 y 3"

Según Hacer

1

Escribir "Elegiste la opción 1"

2:

"Elegiste la opción 1"

3:

Escribir Elegiste la opción 1

De Otro Mod

Escribir "No elegiste la opción 1, ni 2, ni 3"

FinAlgoritmo

¿Cuál es el resultado a lograr?

```
*** Ejecución Iniciada. ***
Ingrese un número entre 1 y 3
> 3
Elegiste la opción 3
*** Ejecución Finalizada. ***
```



Revisemos lo aprendido hasta aquí

Implementación Estructura Según, permitiendo al programa ejecutar un bloque de código según varias condiciones, en vez de tener una sola condición y un bloque de código para el verdadero y el falso, acá podemos tener varias condiciones distintas y un bloque de código para cada opción. Y, además, tenemos el de otro modo para el caso que sean falsas todas las opciones.

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

CONDICIONALES ANIDADOS O EN CASCADA

Es posible también utilizar la instrucción *Si* para diseñar estructuras de selección que contengan más de dos alternativas. Por ejemplo, una estructura *Si-entonces* puede contener otra estructura *Si-entonces*, y esta estructura *Si-entonces* puede contener otra, y así sucesivamente cualquier número de veces; a su vez, dentro de cada estructura pueden existir diferentes acciones, a esto se le llama condicionales anidados o en cascada.

CONDICIONALES ANIDADOS EN PSEINT:

```
Si expresion_logica1 Entonces
    acciones_por_verdadero1
Sino
Si expresion_logica2 Entonces
    acciones_por_verdadero2
Sino
    Si expresion_logica4 Entonces
        acciones_por_verdadero3
    Sino
        acciones_por_falso
Fin Si
Fin Si
Fin Si
```




¿NECESITAS UN EJEMPLO?

```
1  Algoritmo SiAnidado
2  Definir nota Como Entero
3  Escribir "Ingrese su nota"
4  Leer nota
5  /// Anidamos los si para tener una accion para las distintas posibilidades
6  Si nota ≤ 6 Entonces
7      Escribir "Desaprobo"
8  SiNo
9      Si nota = 7
10         Escribir "Aprobo"
11     SiNo
12         Si nota = 8
13             Escribir "Muy bien"
14         SiNo
15             Si nota = 9 Entonces
16                 Escribir "Sobresaliente"
17             SiNo
18                 Si nota = 10
19                     Escribir "Excelente"
20             FinSi
21         FinSi
22     FinSi
23 FinSi
24 Fin Si
25 FinSi
26 FinSi
27
28 FinAlgoritmo
```



Pueden encontrar un ejemplo para descargar de Condicionales Anidados en Moodle.



MANOS A LA OBRA!

EJERCICIO DESAYUNO

Es tu turno, diseña un condicional anidado que le pregunte al usuario si quiere tomar té o café y en caso de que quiera tomar café, preguntar si solo o cortado y en caso de ser cortado, si prefiere leche vegetal.

DETECCIÓN DE ERRORES

Copia y pega este código, que muestra el mayor de 3 numeros en tu programa. Deberás corregir los errores hasta lograr el siguiente resultado esperado:

Algoritmo Correccion_SiAnidado

Definir n1, n2 Como Entero

Escribir "MOSTRAR EL MAYOR DE 3 NÚMEROS"

Escribir "INGRESE NÚMERO 01 : "

n1

Escribir "INGRESE NÚMERO 02 : "

Escribir "INGRESE NÚMERO 03 : "

Leer n3

Si (n1 > n2 n1 n3) Entonces

Escribir "MAYOR ES ", n1

No

Si (n2 n1 n2 n3) Entonces

Escribir "MAYOR ES ", n2

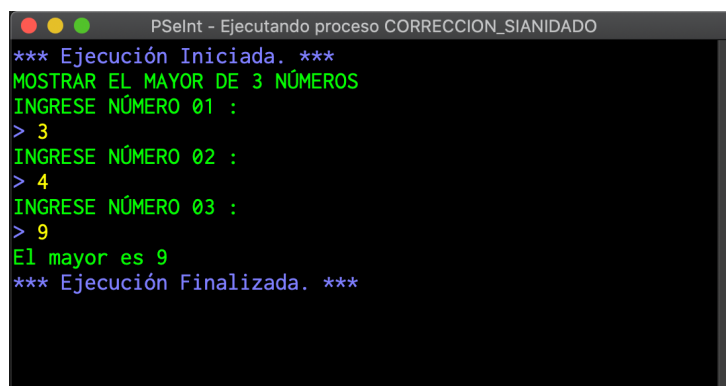
Escribir "MAYOR ES ", n3

FinSi

FinSi

FinAlgoritmo

¿Cuál es el resultado a lograr?



```
*** Ejecución Iniciada. ***
MOSTRAR EL MAYOR DE 3 NÚMEROS
INGRESE NÚMERO 01 :
> 3
INGRESE NÚMERO 02 :
> 4
INGRESE NÚMERO 03 :
> 9
El mayor es 9
*** Ejecución Finalizada. ***
```



Revisemos lo aprendido hasta aquí

- Identificar, diferenciar y elaborar condicionales anidados. Donde el programa puede evaluar una condición y en base a ella, evaluar otras condiciones tanto si son verdaderas o falsas y ejecutar bloques de códigos cuando esas condiciones sean verdaderas.

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.



¿NECESITAS UN EJEMPLO?

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.

PASOS PARA LA CONSTRUCCIÓN DE UN PROGRAMA

Hemos llegado al final de la guía, ahora pondremos en practica todo lo visto a través de una serie de ejercicios, pero, **antes que arranquemos los ejercicios es importante entender los pasos para construir un programa y de esa forma trabajar de la mejor manera posible.**

El proceso de programación es un proceso de solución de problemas en el cual deben llevarse a cabo los pasos descritos a continuación.

¿ CÓMO VAMOS A ENCARAR ESTOS PROBLEMAS ?

Para poder resolver problemas vamos a tener que ejecutar una serie de pasos que nos van a ayudar a resolver el problema sin importar que tan grande o chico sea.

Los pasos serían:

1. Lectura

Leer el problema o la consigna dos veces, la primera para entender de manera general lo que debemos hacer y una segunda vez para entender el problema de manera más concreta y evitar saltarnos algún dato importante.

2. Papel y lápiz

Una vez que tenemos una idea clara de lo que debemos hacer, vamos a dejarlo por escrito, para esto lo mejor es usar papel y lápiz. Vamos a dejar por escrito que debemos realizar. Utilizamos la misma noción que cuando lo leímos, escribimos un esbozo general y luego un esbozo particular.

3. Subproblemas

Cuando estemos haciendo el esbozo particular, pensamos en el concepto divide y vencerás. Pensamos el problema que tenemos y lo dividimos en subproblemas. Un ejemplo sería hacer pan. Hacer pan es un problema que lo podemos dividir en subproblemas. Mezclar la harina con la levadura, amasar el pan, dejarlo levar, etc.

Básicamente, tomamos un problema grande y general para convertirlo en pequeños problemas más concretos y fáciles de afrontar.

4. Herramientas del código

En el siguiente paso, empezamos a pensar en código, antes estábamos tomando un problema de programación y resolviéndolo como si fuera un problema como hacer pan. Ahora vamos a tener que tomar esos subproblemas que habíamos creado y ver que herramientas de programación vamos a necesitar. Esto puede ser variables, constantes, bucles, condicionales, expresiones lógicas o matemáticas, etc. Cualquier herramienta que creamos necesaria para lograr nuestro cometido.

5. Pasaje a código

Y, por último, tomando todo lo que hicimos, lo pasaremos a código. Es importante que cuando empecemos a escribir el código y empecemos a trasladar los subproblemas a código, ir probando que cada subproblema funcione, o cumpla con el resultado esperado. Para facilitar este proceso Pseint, nos presenta la herramienta de ejecutar el código Paso a Paso y la prueba de escritorio.

¿ME BLOQUEE EN CUANTO A CÓDIGO?

- ¡¡Si hay errores rojos!! Leer la descripción del error y la línea donde está el error.
- Indentar el código, para tenerlo bien ordenado. Para indentar, tenemos que seleccionar todo el código, hacer click derecho y click izquierdo en indentar.
- ¿El código hace lo que quiero que haga? En caso que no, correr el código paso a paso y hacer prueba de escritorio.

HICE TODO LO ANTERIOR Y NO PUDE RESOLVERLO

- Primero consulto a mis compañeros como ellos encararon el ejercicio. Les pido que me expliquen en vez de mostrarme lo que hicieron.
- También puedes orientarte en cómo ir resolviendo los subproblemas en internet, es muy difícil encontrar la solución puntual, pero puedes orientarte en buscar parte de la solución
- Si no lo saben tus compañeros ni lo encontraste en internet, consúltale al Mentor Experto de cómo se debería encarar la solución. El Mentor Experto te dará las herramientas necesarias para resolverlo. En el caso de que después de la explicación del Mentor Experto, seguís sin entender, plantea que parte no entendiste.

EJERCICIOS DE APRENDIZAJE

Llegamos hasta acá y adquirimos diversas capacidades para controlar la ejecución de nuestro programa en función de condiciones que nosotros mismos determinamos. Es hora de poner en práctica lo aprendido, comparar con tus compañeros de equipo cómo lo resolvió cada uno y conversar entre ustedes aquellos conceptos que no hayan quedado del todo claros. Recuerda que, si no logras resolver un ejercicio, debes avisarle al Facilitador de tu equipo para que te ayude, o haga que el resto del equipo trabaje en conjunto para explicarte. Si aún con la ayuda, no pueden resolverlo, llamen a un Mentor con el botón de ayuda.



VIDEOS: Te sugerimos ver los videos relacionados con este tema, antes de empezar los ejercicios, los podrás encontrar en tu aula virtual o en nuestro canal de YouTube.

Condición Simple

1. Un hombre desea saber si su sueldo es mayor al sueldo mínimo, el programa le pedirá al usuario su sueldo actual y el sueldo mínimo. Si el sueldo es mayor al mínimo se debe mostrar un mensaje por pantalla indicándolo.

Condicional Doble

2. Realiza un programa que sólo permita introducir los caracteres 'S' y 'N'. Si el usuario ingresa alguno de esos dos caracteres se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO".
3. Realizar un programa que pida un número y determine si ese número es par o impar. Mostrar en pantalla un mensaje que indique si el número es par o impar. (para que un número sea par, se debe dividir entre dos y su resto debe ser igual a 0). **Nota: investigar la función mod de PseInt.**
4. Realizar un programa que pida introducir solo frases o palabras de 6 caracteres. Si el usuario ingresa una frase o palabra de 6 caracteres se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO". **Nota: investigar la función Longitud() de PseInt.**
5. Realizar un programa que pida una frase o palabra y si la frase o palabra es de 4 caracteres de largo, el programa le concatenara un signo de exclamación al final, y si no es de 4 caracteres el programa le concatenara un signo de interrogación al final. El programa mostrará después la frase final. **Nota: investigar la función Longitud() y Concatenar() de PseInt.**
6. Escriba un programa que pida 3 notas y valide si esas notas están entre 1 y 10. Si están entre esos parámetros se debe poner en verdadero una variable de tipo lógico y si no ponerla en falso. Al final el programa debe decir si las 3 notas son correctas usando la variable de tipo lógico.
7. Escriba un programa que pida una frase o palabra y valide si la primera letra de esa frase es una 'A'. Si la primera letra es una 'A', se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO". **Nota: investigar la función Subcadena de PseInt.**
8. Continuando el ejercicio anterior, ahora se pedirá una frase o palabra y se validara si la primera letra de la frase es igual a la última letra de la frase. Se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO".

Condicional Múltiple

9. Construir un programa que simule un menú de opciones para realizar las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división) con dos valores numéricos enteros. El usuario, además, debe especificar la operación con el primer carácter de la operación que desea realizar: 'S' o 's' para la suma, 'R' o 'r' para la resta, 'M' o 'm' para la multiplicación y 'D' o 'd' para la división.

Condicionales Anidados

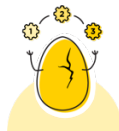
10. Realizar un programa que, dado un número entero, visualice en pantalla si es par o impar. En caso de que el valor ingresado sea 0, se debe mostrar "el número no es par ni impar".

Nota: investigar la función mod de PSeInt

11. Escriba un programa para obtener el grado de eficiencia de un operario de una fábrica de tornillos, de acuerdo a las siguientes dos condiciones que se le imponen para un período de prueba:

- Producir *menos* de 200 tornillos defectuosos.
- Producir *más* de 10000 tornillos sin defectos.
- El grado de eficiencia se determina de la siguiente manera:
- Si no cumple ninguna de las condiciones, grado 5.
- Si sólo cumple la primera condición, grado 6.
- Si sólo cumple la segunda condición, grado 7.
- Si cumple las dos condiciones, grado 8

Nota: para trabajar este ejercicio de manera prolija, ir probando cada inciso que pide el ejercicio. No hacer todos al mismo tiempo y después probar.



¿Lograste los objetivos de la guía?

¡Terminamos los ejercicios fundamentales de esta guía! **Te invitamos a ir a tu aula virtual y contestar una breve encuesta para revisar los objetivos de la esta guía**, reflexiona sobre cada uno de ellos haciéndote la siguiente pregunta ¿Pude aplicar este objetivo al realizar los ejercicios? Si es así, ¡Excelente!

¿Terminaste con los ejercicios fundamentales? hay más ejercicios para que puedas seguir practicando.

¿No pudiste cumplir los objetivos de la guía?

Si aún no has logrado tildar los objetivos propuestos, **no te preocupes**, a través de Slack contacta a un Coach para que te ayude a diseñar una estrategia personalizada para entender qué pasó y poder avanzar.

EJERCICIOS DE APRENDIZAJE EXTRA

Estos van a ser ejercicios para reforzar los conocimientos previamente vistos. Estos pueden realizarse cuando hayas terminado la guía y tengas una buena base sobre lo que venimos trabajando. Además, si ya terminaste la guía y te queda tiempo libre en las mesas, puedes continuar con estos ejercicios extra, recordando siempre que no es necesario que los termines para continuar con el tema siguiente. Por último, recordá que la prioridad es ayudar a los compañeros de la mesa y que cuando tengas que ayudar, lo más valioso es que puedas explicar el ejercicio con la intención de que tu compañero lo comprenda, y no sólo mostrarlo. ¡Muchas gracias!

Condicional Doble

1. Realizar un programa que pida tres notas y determine si un alumno aprueba o reprueba un curso, sabiendo que aprobará el curso si su promedio de tres calificaciones es mayor o igual a 70; y reprueba en caso contrario.
2. Una tienda ofrece para los meses de septiembre, octubre y noviembre un descuento del 10% sobre el total de la compra que realiza un cliente. Solicitar al usuario que ingrese un mes y el importe de la compra. El programa debe calcular cuál es el monto total que se debe cobrar al cliente e imprimirlo por pantalla.
3. Solicitar al usuario que ingrese dos números enteros y determinar si ambos son pares o impares. Mostrar en pantalla un mensaje que indique “Ambos números son pares” siempre y cuando cumplan con la condición. En caso contrario se deberá imprimir el siguiente mensaje “Los números no son pares, o uno de ellos no es par”.

Nota: investigar la función mod de PseInt.

4. La empresa “Te llevo a todos lados” está destinada al alquiler de autos y tiene un sistema de tarifa que consiste en cobrar el alquiler por hora. Si el cliente devuelve el auto dentro de las 2 horas de uso el valor que corresponde pagar es de \$400 pesos y la nafta va de regalo. Cuando el cliente regresa a la empresa pasadas las 2 horas, se ingresan la cantidad de litros de nafta gastados y el tiempo transcurrido en horas. Luego, se le cobra 40 pesos por litro de nafta gastado, y la hora se fracciona en minutos, cobrando un total de \$5,20 el minuto de uso. Realice un programa que permita registrar esa información y el total a pagar por el cliente.

Condicional Múltiple

5. Solicitar al usuario que ingrese un valor entre 1 y 7. EL programa debe mostrar por pantalla un mensaje que indique a qué día de la semana corresponde. Considere que el número 1 corresponde al día “Lunes”, y así sucesivamente.
6. Leer tres números que denoten una fecha (día, mes, año) y comprobar que sea una fecha válida. Si la fecha no es válida escribir un mensaje de error por pantalla. Si la fecha es válida se debe imprimir la fecha cambiando el número que representa el mes por su nombre. Por ejemplo: si se introduce 1 2 2006, se deberá imprimir “1 de febrero de 2006”.
7. Hacer un algoritmo que lea un número por el teclado y determine si tiene tres dígitos.

Condicionales Anidados

8. Si se compran menos de cinco llantas el precio es de \$3000 cada una, si se compran entre 5 y 10 el precio es de \$2500, y si se compran más de 10 el precio es \$2000. Obtener la cantidad de dinero que una persona tiene que pagar por cada una de las llantas que compra, y el monto total que tiene que pagar por el total de la compra.
9. Realice un programa que, dado un año, nos diga si es bisiesto o no. Un año es bisiesto bajo las siguientes condiciones: Un año divisible por 4 es bisiesto y no debe ser divisible por 100. Si un año es divisible por 100 y además es divisible por 400, también resulta bisiesto. **Nota: recuerde la función mod de PseInt**
10. Una verdulería ofrece las manzanas con descuento según la siguiente tabla:

Nº DE KILOS COMPRADOS	% DESCUENTO
0 – 2	0%
2.01 – 5	10%
5.01 – 10	15%
10.01 en adelante	20%

Determinar *cuánto pagará* una persona que compre manzanas en esa verdulería

11. El promedio de los trabajos prácticos de un curso se calcula en base a cuatro notas de las cuales se elimina la nota menor y se promedian las tres notas más altas. Escriba un programa que determine cuál es la nota eliminada y el promedio de los trabajos prácticos de un estudiante.
12. Una empresa tiene personal de distintas áreas con distintas condiciones de contratación y formas de pago. El departamento de contabilidad necesita calcular los sueldos semanales (lunes a viernes) en base a las 3 modalidades de sueldo:
- a) comisión
 - b) salario fijo + comisión, y
 - c) salario fijo
- a) Para la modalidad salario por comisión se debe ingresar el monto total de las ventas realizadas en la semana, y el 40% de ese monto total corresponde al salario del empleado.
- b) Para la condición de salario fijo + comisión, se debe ingresar el valor que se paga por hora, la cantidad de horas trabajadas semanalmente y el monto total de las ventas en esa semana. En este tipo de contrato las horas extras no están contempladas y se fija como máximo 40 horas por semana. La comisión por las ventas se calcula como 25% del valor de venta total.
- c) Finalmente, para la modalidad de salario fijo se debe ingresar el valor que se paga por hora y la cantidad de horas trabajadas en la semana. En el caso de exceder las 40 horas semanales, las horas extras se deben pagar con un extra del 50% del valor de la hora. Realizar un menú de opciones para poder elegir el tipo de contrato que tiene un empleado.