



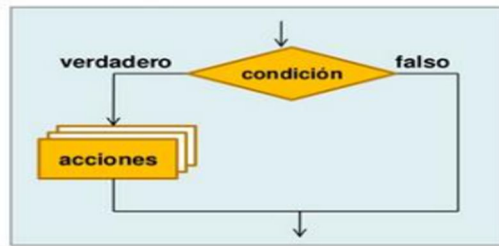
PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL FORMATO GUÍA DE APRENDIZAJE

- **DENOMINACIÓN DEL PROGRAMA DE FORMACIÓN:** Análisis y desarrollo de software
- **CÓDIGO DEL PROGRAMA DE FORMACIÓN:** 2615103
- **NOMBRE DEL PROYECTO:**
- **FASE DEL PROYECTO:** Análisis
- **ACTIVIDAD DE PROYECTO:** Análisis y comprensión de herramientas tecnológicas necesarias para el manejo de la información en la empresa
- **COMPETENCIA** Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.
- **RESULTADOS DE APRENDIZAJE ALCANZAR:** Valorar la incidencia de los datos en los procesos del macrosistema, tomando como referente el diccionario de datos y las mini especificaciones, para la consolidación de los datos que intervienen, de acuerdo con parámetros establecidos.
- **DURACIÓN DE LA GUÍA:** 18 horas

2. Presentación

En esta etapa iniciamos el trabajo y manejo de estructuras condicionales y de repetición y son utilizadas de manera muy variada, dependiendo de la problemática a resolver del programador deberá aplicar la más pertinente. Se busca desarrollar habilidades que permitan implementar diversa estructura selectiva de forma adecuada en la solución de un problema, utilizando la herramienta de pseudocódigo. En programación, una sentencia condicional es una instrucción o grupo de instrucciones que se pueden ejecutar o no en función del valor de una condición. Los tipos más conocidos de sentencias condicionales son el SI..ENTONCES (if..then), el SI..ENTONCES..SINO (if..then..else).SI. ENTONCES: Es un estamento que se utiliza para probar si una determinada condición se ha alcanzado, como por ejemplo averiguar si un valor analógico está por encima de un cierto número, y ejecutar una serie de declaraciones (operaciones) que se escriben dentro de llaves, si es verdad. Si es falso (la condición no se cumple) el programa salta y no ejecuta las operaciones que están dentro de las llaves.

Anteriormente hemos estado haciendo programas que solo hacen cálculos, pero la programación es más interesante cuando nuestros programas toman sus propias decisiones. Las estructuras condicionales comparan una variable contra otro(s) valor(es) mediante un operador lógico o relacional, para que en base al resultado de esta comparación, se siga un curso (tome una decisión) de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Se clasifican en:



- **Simples:** Las estructuras condicionales simples se les conoce como “Tomas de decisión”. Estas tomas de decisión tienen la siguiente forma:



- **Dobles:** Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición. Se representa de la siguiente forma:

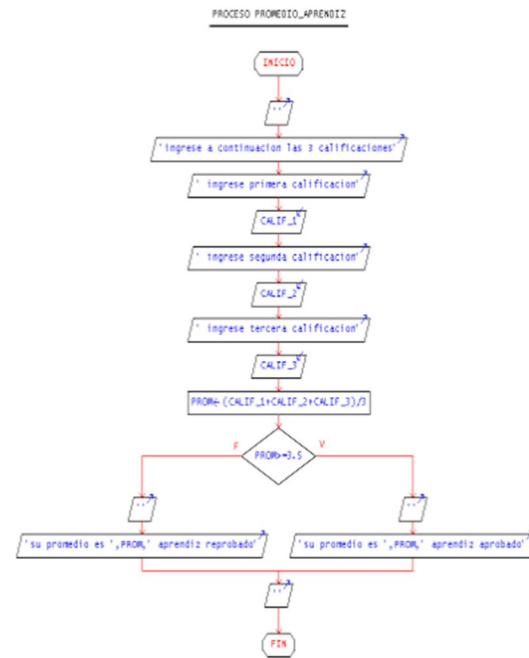


Veamos un Ejemplo

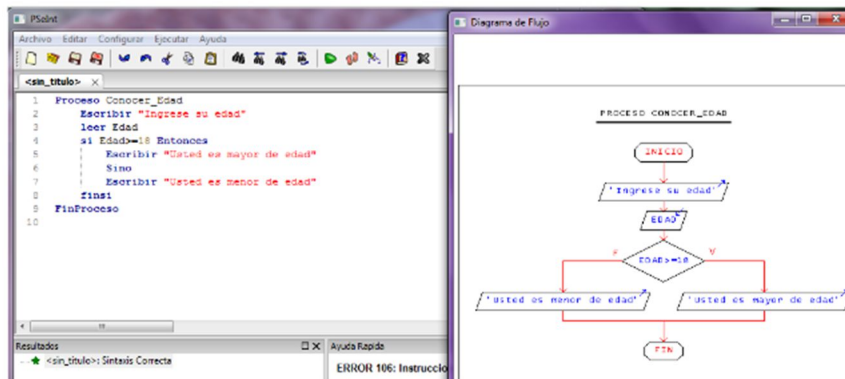
1 Determinar si un aprendiz aprueba o reprueba un curso, sabiendo que aprobará si su promedio de tres calificaciones es mayor o igual a 3.5; reprueba en caso contrario.



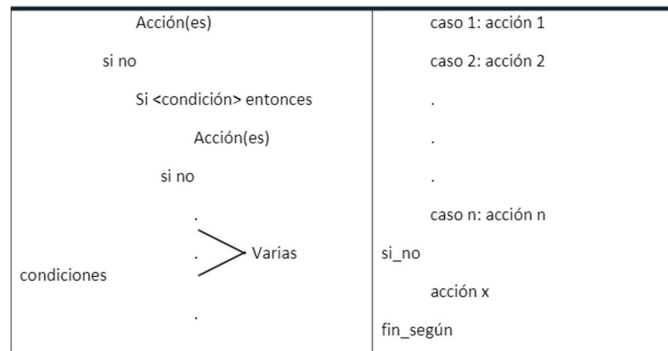
```
1 // Determinar si un aprendiz aprueba o reprueba un curso, sabiendo que
2 // aprobará si su promedio de tres calificaciones es mayor o igual a 70; reprueba en caso contrario.
3 Proceso promedio_aprendiz
4 escribir ""
5 Escribir "ingrese a continuación las 3 calificaciones"
6 Escribir "ingrese primera calificación"
7 leer calif_1
8
9 Escribir "ingrese segunda calificación"
10 leer calif_2
11
12 Escribir "ingrese tercera calificación"
13 leer calif_3
14
15 prom <- (calif_1 + calif_2 + calif_3)/3
16
17 Si prom >= 70 entonces
18     escribir ""
19     escribir "su promedio es ", prom, " aprendiz aprobado"
20 Sino
21     escribir ""
22     escribir "su promedio es ", prom, " aprendiz reprobado"
23 FinSi
24
25 escribir ""
26 FinProceso
27
```



2 Ejemplo: Determinar si una persona es mayor en base a la edad ingresada.



Múltiples: Las estructuras de comparación múltiples, son tomas de decisión especializada que permiten comparar una variable o expresión contra distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma común es la siguiente

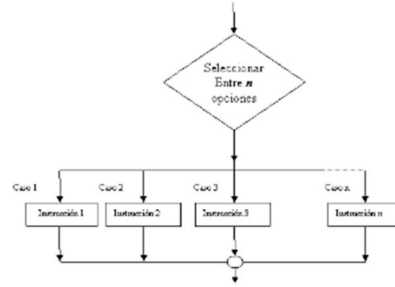


Veamos un ejemplo

Indicar a que mes corresponde el número ingresado, tomando como valores validos los números del 1-12, en caso contrario indicar que el número es inválido

```
Proceso Meses
  Escribir "Por favor ingrese un numero entre 0 y 12"
  Leer mes
  Segun mes Hacer
    1: Escribir "El mes es Enero"
    2: Escribir "El mes es Febrero"
    3: Escribir "El mes es Marzo"
    4: Escribir "El mes es Abril"
    5: Escribir "el mes es Mayo "
    6: Escribir "el mes es Junio"
    7: Escribir "el mes es Julio"
    8: Escribir "el mes es Agosto"
    9: Escribir "el mes es Septiembre"
    10: Escribir "el mes es Octubre"
    11: Escribir "el mes es Noviembre"
    12: Escribir "el mes es Diciembre "
  De Otro Modo:
    Escribir "No corresponde a ningun mes"
  FinSegun
FinProceso
```

La estructura de selección múltiple ofrece una solución más simple y compacta. Para esta estructura se evalúa una expresión que puede tomar n valores distintos (1, 2, 3, ..., n) a cada valor de la expresión corresponde una instrucción (o serie de instrucciones) diferente, lo que significa que el flujo seguirá un camino distinto entre los n posibles caminos en dependencia del valor de la expresión de control. La representación gráfica de este tipo de estructura de selección sería:



ACTIVIDADES PROPUESTAS

Estimado Aprendiz, realice el pseudocódigo en Pseint guarde en carpeta:

📁 Semana 2

📁 1. Pseint_Estructura_Control

1. Escribir un programa que a partir de un número ingresado diga si es mayor, menor o igual a 9.
2. Crear un programa que a partir de un número ingresado diga si el mismo es par o impar.
3. Escribir un programa que permita ingresar un número y muestre en pantalla todos los divisores del mismo.
4. Determinar si un alumno aprueba o reprueba un curso, sabiendo que aprobará si su promedio de tres calificaciones es mayor o igual a 4.0; reprueba en caso contrario. Deberá permitir ingresar las tres calificaciones y luego calcular su promedio.
5. Crear un programa que permita ingresar un nombre y una cantidad numérica para que así después el programa escriba este nombre tantas veces como su cantidad ingresada. Utilizar la función MIENTRAS.
6. Escribir un programa que permita sumar todos los números naturales comprendidos entre 1 y 50 utilizando la función REPETIR.
7. Crear un programa que efectúe la suma de los dígitos de un número ingresado. Ejemplo: Si se ingresa 123, debería devolver 6.



8. **Ejercicio suma:** Se desea obtener la suma de dos números cualesquiera, si el resultado es mayor a 5.
9. **Ejercicio Productor de Leche:** Un programa que pide al usuario ingresar la cantidad de leche producida en litros por un productor de leche y que calcula la cantidad de dinero que recibirá el productor al vender la leche. El programa debe tener en cuenta que cuando el productor entrega la leche, le pagan en galones, y que el precio de venta de un galón de leche es de \$3.75. Además, el programa debe validar que la cantidad de litros ingresada sea mayor que cero.
10. **Ejercicio Edad:** Un programa que pide al usuario ingresar el año de nacimiento de una persona y que calcula su edad en años. El programa debe tener en cuenta que la edad no puede ser negativa ni mayor a 120 años.
11. **Ejercicio Parqueadero:** Un programa que pide al usuario ingresar la cantidad de horas que ha utilizado un estacionamiento y que calcula el cobro que debe aplicarse. El programa debe tener en cuenta que las fracciones de hora se toman como completas y que el cobro es de \$10 por hora o fracción de hora.
12. **Ejercicio la Brocha Gorda:** Un programa que pide al usuario ingresar la cantidad de metros cuadrados a pintar y que calcule el costo total del trabajo de pintura. El programa debe tener en cuenta que se cobra por metro cuadrado y que el costo por metro cuadrado es de \$100.
13. **Valor llamada:** Determinar la cantidad total a pagar de una llamada telefónica, teniendo en cuenta lo siguiente:
 - Toda llamada que dure 3 minutos tiene un costo de 200 pesos por minuto
 - Toda llamada que dure entre 3 y 10 minutos cuesta 150 pesos por minuto
 - Toda llamada que dure más de 10 minutos cuesta 100 pesos por minuto
14. **Ejercicio Promedio:** Un estudiante realiza cuatro exámenes durante el semestre, calcular el promedio. Siguiendo el mismo problema ahora agregaremos una condición (comando si-entonces-sino), para enviar un mensaje de “Aprobado” o “Reprobado”. En el caso de que el promedio fuera menor a 6 se imprimiría al final “REPROBADO”. Lo guardaremos como: `Calcular_promedio2`
15. Bien, ahora realizaremos el mismo ejercicio, pero para N alumnos, como N es un valor desconocido tendremos que preguntar al usuario que valor desea darle a N y además utilizaremos el ciclo mientras. A este ejercicio le nombraremos `Calcula_promedio3`
16. Para terminar, realizaremos un menú de tres opciones con el comando -según-, en cada una de las opciones de colocaremos cada uno de los ejercicios realizados con anterioridad. Le nombraremos menú. A este ejercicio lo guardaremos `Calcula_promedio_opciones`.



Escribir "MENU DE OPCIONES";
Escribir "1. CALCULA PROMEDIO";
Escribir "2. CALCULA PROMEDIO E IMPRIME APROBADO O REPROBADO";
Escribir "3. PARA N ALUMNOS CALCULA PROMEDIO E IMPRIME APROBADO O REPROBADO";
Escribir "0. PARA SALIR DIGITA CERO";
Escribir "Introduce un numero de opción";

17. Ejercicio Tipo de Triángulo: Un programa que determina el tipo de triángulo según las medidas de sus lados. El programa debe pedir al usuario ingresar la medida de cada uno de los tres lados del triángulo y luego imprimir un mensaje indicando si es equilátero, isósceles o escaleno.

- Triángulo equilátero: Las medidas de sus tres lados son iguales.
- Triángulo isósceles: Las medidas de dos lados son iguales, es decir, dos lados son congruentes.
- Triángulo escaleno: Todas las medidas de sus lados son diferentes, es decir, no tiene lados congruentes.

Pista : En este ejemplo, se utilizan dos estructuras de control de condición "Si" y "SiNo" para determinar si el triángulo es equilátero, isósceles o escaleno.

18. Ejercicio Selecciona operación: Un programa que solicita al usuario ingresar dos números enteros y realiza una operación aritmética según la opción que elija el usuario: suma, resta, multiplicación o división. El programa debe verificar que la opción ingresada sea válida y, en caso contrario, debe imprimir un mensaje indicando que la opción es inválida.

Pista: En este ejemplo, se utiliza una estructura de control "Si" para comprobar si la opción ingresada es válida y, si es así, se utiliza otra estructura de control "Si" para realizar la operación aritmética correspondiente. Si la opción ingresada es inválida, el programa imprimirá un mensaje indicando que la opción es inválida.