

**INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS  
AVANZADAS**

## **Práctica No. 2. Manejo de Sentencias de Control.**

**Unidad Temática:** II Estructuras de control y Aplicaciones con Arreglos.

**Lugar de realización:** Laboratorio de Cómputo

**Duración:** 12 hrs.

### **Objetivo**

Desarrollo de programas utilizando las sentencias de control

### **Resultados Esperados**

- Identificación de elementos del diagrama de flujo logrado en el pre-reporte
- Transpolar diagramas de flujo a código fuente.
- Construcción de programas en Lenguaje C para conocer la diferencia del uso de sentencias de control
- Identificación y selección de sentencias de control para procesos repetitivos



### **Actividad previa.**

Para el buen desarrollo de la práctica previamente el alumno entregará un memorama de 10 pares de diagramas de flujo. Un **memorama** es un juego de mesa que trata de encontrar cartas parejas, en este caso la pareja estará compuesta por:

1. Una carta que tendrá una sección de un diagrama de flujo y
2. Su pareja tendrá el código fuente correspondiente

Para ello es necesario que el alumno conozca los diagramas de flujo previamente vistos en clase.

*“El fin se encuentra en el inicio”*

## Material y Equipo.

- IDE Dev-Cpp o compatible.
- Computadora

## Introducción.

### Sentencias de Control

Las sentencias de control que existen son:

- Secuencial
- Selección
- Repetición

Las sentencias de selección nos permiten ejecutar cierta parte del código siempre y cuando se cumpla una condición, existen dos estructuras de selección en C/C++: `if` y `switch-case`.

### Sentencia IF

Si la condición evaluada en la sentencia `if` es verdadera se ejecuta la acción<sub>1</sub> y si es falsa se ejecuta la acción<sub>2</sub>

Sintaxis:

```
if (condición){
    acción 1;
}else{
    acción 2;
}
```

### Ejemplo:

```
if(n%d == 0)
    printf("%d" es divisible entre %d \n", n, d);
else
    printf("%d" no es divisible entre %d \n", n, d);
```

Las sentencias `if-else` anidadas permiten implementar decisiones que impliquen diferentes opciones.

### Ejemplo:

```
if(x > 0){
    z = 2*log(x);
}else if(y > 0){
    z = sqrt(y);
}else
    printf("Imposible calcular z");
```

### Sentencia SWITCH

La **sentencia switch**: Es una sentencia de control útil para seleccionar una de entre múltiples alternativas. El selector puede ser de tipo int o char.

Sintaxis:

```
switch (variable){
    case valor1:
        acción 1;
        break;
    case valor2:
        acción 2;
        break;
    . . .
    default:
        acción n;
        break;
}
```

Ejemplo:

```
switch(nota){
    case 10: printf("Felicidades, examen superado"); break;
    case 9:  printf("Notable"); break;
    case 8:  printf("Aprobado"); break;
    default: printf("Necesitas estudiar más"); break;
}
```

Las *sentencias de repetición* nos permiten ejecutar cierta parte del código en repeticiones.

### Sentencia WHILE

La **sentencia while**: Repite una sentencia o sentencias n veces, siempre y cuando se cumpla la condición.

Sintaxis:

```
while (condición){
    acción 1;
    . . .
}
```

Ejemplo:

```
int x = 0;
while(x < 10)
    printf("x: %d \n", ++x);
```

### Sentencia FOR

La **sentencia for** sirve para ejecutar un bloque de sentencias un número fijo de veces.

Sintaxis:

```
for(valor_inicial;condición;incremento){
    acciones;
}
```

Ejemplo:

```
for(i = 0; i < 10; i++)
    printf("Hola\n");
```

### Sentencia DO-WHILE

La **sentencia do-while**: Se utiliza para especificar un bucle condicional que se ejecuta al menos una vez.

Sintaxis:

```
do{
    acciones;
}while(condición);
```

Ejemplo:

```
do{
    printf("<<Introduzca un
    digito (0-9)");
    scanf("%d", &num);
}while(num >= 0 && num <= 9);
```

### Desarrollo.

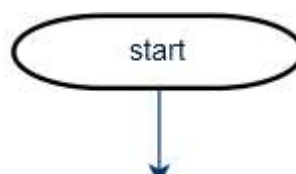
#### Ejemplo:

Se requiere saber el estado del tanque de gasolina de un automóvil. El usuario ingresará la capacidad del tanque de gasolina de su auto y la cantidad de litros que cargó en la gasolinera cuando su tanque estaba vacío. Dibuje el Diagrama de Flujo y escriba un programa que despliegue TANQUE VACÍO en caso de estar en menos del 30%, desplegará LLENO en caso de estar a más del 80% y mostrará MEDIO en cualquier otro caso.

#### Solución:

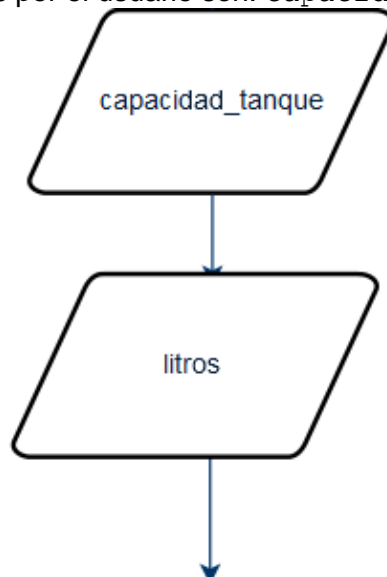
**Diagrama de flujo** Ver *diagrama 1gasolina.dfd* anexo en disco

1. Inicio del programa:



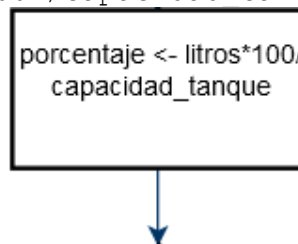
## 2. Identificación de entradas:

Las entradas ingresadas por el usuario son: `capacidad_tanque` y `litros`



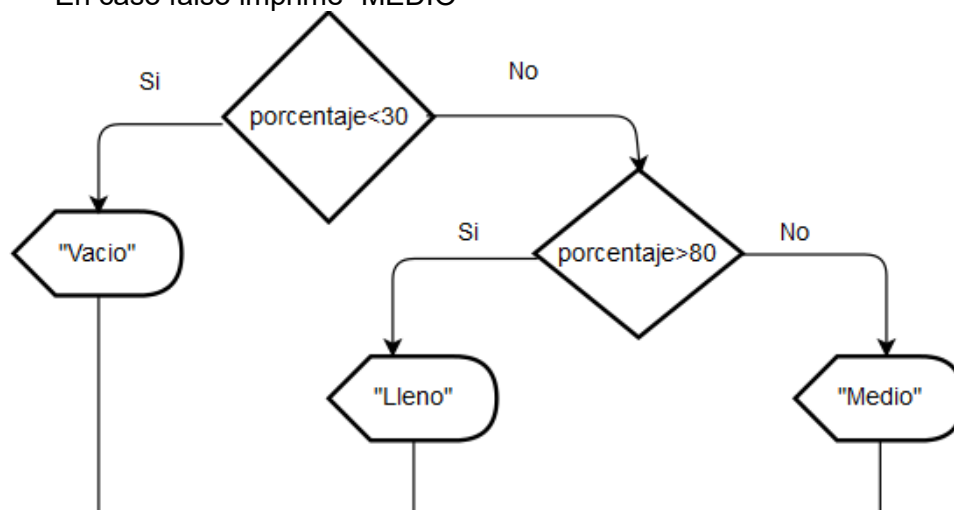
## 3. Realizar cálculo:

$\text{porcentaje} = \text{litros} * 100 / \text{capacidad\_tanque}$



## 4. Realizar condición:

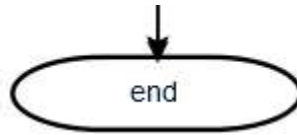
La primera condición es si es menor a 30  
En caso verdadero imprime "VACÍO"  
En caso falso pregunta si es mayor a 80  
En caso verdadero imprime "LLENO"  
En caso falso imprime "MEDIO"



Identificación de salida:

La salida será: LLENO, VACÍO o MEDIO dependiendo del porcentaje de llenado.

5. Fin



### Programa en C *Ver código 2gasolina.c anexo en disco*

1. Inicio del programa:

```
#include <stdio.h>
int main() {
```

2. Identificación de entradas:

Se debe de identificar el tipo de dato adecuado para declarar las variables.

```
float capacidad_tanque, litros, porcentaje;
```

Se solicitan los datos al usuario:

```
scanf("%f", &capacidad_tanque);
scanf("%f", &litros);
```

3. Realizar cálculo:

```
porcentaje = litros*100 / capacidad_tanque;
```

4. Realizar condición:

```
if (porcentaje < 30) {           //La primera condición es si es menor a 30
    printf("VACÍO");             //En caso verdadero imprime "VACÍO"
} else {                         //En caso falso
    if (porcentaje > 80) { //pregunta si es mayor a 80
        printf("LLENO"); //En caso verdadero imprime "LLENO"
    } else {
        printf("MEDIO"); //En caso falso imprime "MEDIO"
    }
}
}
```

5. Cerramos llave para terminar el programa

```
}
```

## Programas

Realice los siguientes ejercicios:

1. Dada la función  $f(x)$ , dibujar el diagrama de flujo y el código en C para calcular la función para un valor de  $x$  introducido por el teclado y visualizarlo en pantalla

$$f(x) = \begin{cases} x^2 - x & \text{para } x \leq 0 \\ -x^2 + 3x & \text{para } x > 0 \end{cases}$$

Entrada:

Valor de la variable:  $x$

Salida:

Suponga los valores de entrada  $x=3$ ,  $y=2$ ,  $a=7$  y  $b=4$ , la salida sería:  
res=75

**Procedimiento:**

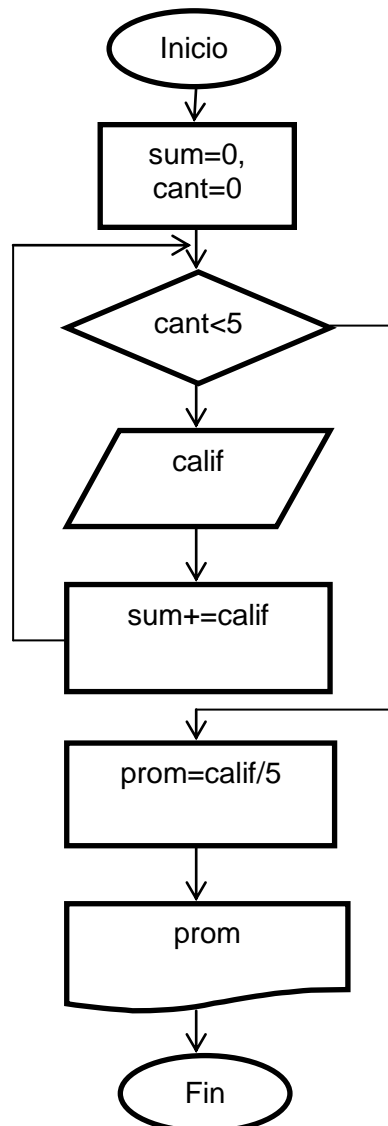
- 1) Declarar  $x$  y  $f_x$
  - 2) Si  $x$  es menor o igual a 0 (hacer condición con sentencia de control correspondiente; considere que tiene caso positivo y caso negativo)
    - a. Caso verdadero: evaluar  $x^2 - x$
    - b. Caso falso: evaluar  $-x^2 + 3x$
  - 3) Imprimir  $f_x$
2. Dadas 5 calificaciones de un alumno; imprima el promedio y la palabra “aprobado” si el alumno tiene un promedio mayor o igual que 6, o “no aprobado” en caso contrario. Escriba el código correspondiente según el diagrama de flujo siguiente:

**Entrada:**

Valor de las calificaciones

**Salida:**

Suponga los valores de entrada 7,9,8,10,6, la salida sería: prom=8

**Procedimiento:**

3. El cobro de luz del actual corresponde a la siguiente tabla según la cantidad de KWH consumidos. Se desea hacer una calculadora de tarifa cuando el usuario ingrese la cantidad de KWH consumidos. Dibuje el diagrama de flujo y escriba el programa correspondiente.

Cargo (\$/KWH)	Por energía	
Rango		Costo
1-50		2.288
51-100		2.762
Adicional		3.042
Cargo Mensual	Fijo	52.84

Entrada:

Valor de la variable: `kwh`

Salida:

Si se consume: 115 KWH entonces:

Cuota fija:	\$ 52.84	
Los primeros 50:	\$ 114.40	quedan 65
Del 51 al 100:	\$ 138.10	quedan 15
Los 15 adicionales:	\$ 45.63	
Total:	<u>\$ 350.97</u>	

Procedimiento:

- 1) Declarar `kwh` y `costo`
- 2) Iniciar `costo` en 0
- 3) Solicitar al usuario el valor de `kwh`
- 4) Si `kwh` es menor o igual a 50
  - a. Caso verdadero: multiplicar `kwh` por 2.288
  - b. Caso falso: Si `kwh` es menor a 100
    - i. Caso verdadero: multiplicar 50 por 2.288 y añadir el sobrante de la resta de 50 multiplicada por 2.762
    - ii. Caso falso: *Determine los siguientes casos*
- 5) Imprimir `kwh`

4. Dibuje el diagrama de flujo correspondiente para este ejercicio. Escriba el código que imprima el siguiente menú:

MENU:

-----

1. Hamburguesa chica con papas y refresco	\$20
2. Hotdog y refresco	\$18
3. Ensalada rusa	\$15

El cliente seleccionará un producto, cuantos productos quiere y genere el total a pagar.

Entrada:

Valor de la variable: `num_productos`, `seleccion`

Salida:

Si se piden 3 productos y selecciona una vez cada uno, el total sería: \$53



**Procedimiento:**

- 1) Declarar `num_productos`, `seleccion` y `total`
- 2) Iniciar `total` en 0
- 3) Solicitar al usuario el valor de `num_productos`
- 4) Repetir `num_productos` veces (seleccionar la sentencia de control que repita determinada cantidad de veces un conjunto de sentencias)
  - a. Pedir el producto y guardarlo en variable `seleccion`.
    - i. En caso de que `seleccion` sea 1
      1. Aumentar `total` en 20
    - ii. En caso de que `seleccion` sea 2
      1. Aumentar `total` en 18
    - iii. En caso de que `seleccion` sea 3
      1. Aumentar `total` en 15
- 5) Imprimir `total`

5. Escribir un programa que simule a una calculadora simple. Lee dos enteros y un carácter. Si el carácter es un +, se imprime la suma, si es un -, se imprime la diferencia, si es un \*, se imprime el producto, si es un /, se imprime el cociente. La calculadora se repite hasta que el usuario escriba 'S' o 's' para salir. Utilizar la sentencia `switch`.

**Entrada:**

Valor de la variable: `num1`, `num2`, `operador`

**Procedimiento:**

- 1) Declarar `num1`, `num2`, `operador`, `total`
- 2) Solicitar al usuario el valor de `num1`, `num2`
- 3) Pedir el `operador`.
  - a. En caso de que `operador` sea '+' entonces `total` es la suma de `num1` y `num2`
  - b. En caso de que `operador` sea '-' entonces `total` es la resta de `num1` y `num2`
  - c. En caso de que `operador` sea '\*' entonces `total` es la multiplicación de `num1` y `num2`
  - d. En caso de que `operador` sea '/' entonces `total` es la división de `num1` y `num2`
- 4) Imprimir `total`
- 5) Preguntar si desea salir. En caso de que sea 'S' o 's' entonces termina, si no regresar a 2.

6. Escribir un programa que presente los valores de la función  $\text{seno}(2x) \cdot x$  para 0, 0.5, 1.0, ..., 9.5, 10. Se puede apoyar del siguiente diagrama de flujo.

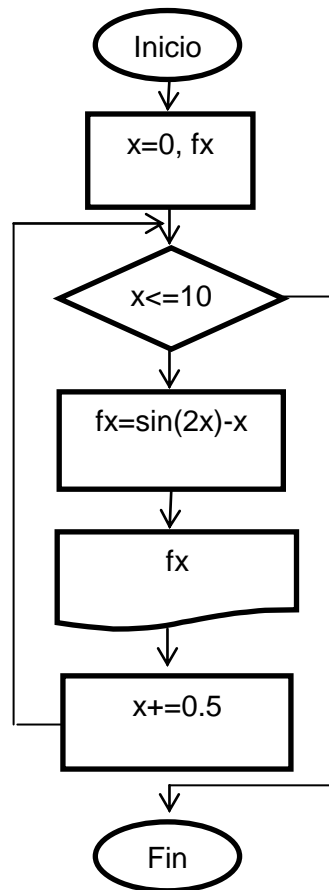
**Entrada:**

-----

**Salida:**

`fx`

Procedimiento:



### Problema de Aplicación

Haciendo uso de sentencias de control, analice, diseñe e implemente el siguiente problema:

**Control de humedad en invernadero.** En un invernadero la humedad relativa (HR) del aire es un factor climático que puede modificar el rendimiento final de los cultivos. Cuando la HR es excesiva las plantas reducen la transpiración y disminuyen su crecimiento.

Suponga un invernadero en el que se suelen cosechar: tomate (con un HR óptimo de 50-60%), melón (entre 60-70%) o pepino (entre el 70-90%), dichos cultivos podrían estar aunados a la siembra de calabacita (entre 55-80%).

Se cuenta con un sensor de humedad (el usuario lo ingresa para efectos de simulación) que monitorea cada 10 segundos en caso de que no exista alguna afectación a los cultivos, si la humedad es mayor se abren las ventilas por 5 segundos (impresión en pantalla para efectos de simulación) provocando un descenso del 3% sobre el valor actual, si la HR es menor se hace un riego por 3 segundos que sube la humedad 4% sobre el valor actual.

Por ejemplo, si se estuviera cosechando solo pepino y la humedad es 95%, se resta el 3% de 95 quedando en 92.15% la HR, como aún está por arriba de los niveles se resta otro 3% sobre 92.15% para obtener una HR de 89.39% que al estar en niveles normales regresa a evaluarse cada 10 segundos.

**Sugerencias en la Solución.**

- Diseñe la solución usando un diagrama de flujo
- Detecte las posibles combinaciones de cosechas
- Simule el paso de tiempo usando la función `sleep`
- Para efectos de la simulación detenga el sistema automáticamente al minuto

**Proyectos (opcionales).**

Codificar un programa que lea 3 números enteros e imprima el mayor de ellos.

Codificar un programa que imprima todos los caracteres ASCII usando un ciclo `for`

Codificar un programa que imprima la tabla de multiplicar del 1 al 9, usando la sentencia de control `for` de manera anidada.

**Evaluación de la Práctica.**

Sección	Elemento a Evaluar
Pre-reporte	Relación diagrama de flujo con código
Ejercicio 1	Sentencia de control <code>if</code> y diagrama de flujo
Ejercicio 2	Transformar diagrama de flujo a código. Selección de ciclo.
Ejercicio 3	<code>if</code> anidados y diagrama de flujo.
Ejercicio 4	Ciclo <code>for</code> y <code>switch</code>
Ejercicio 5	Ciclo <code>while</code> y <code>switch</code>
Ejercicio 6	Ciclo <code>for</code>
Problema	Interpretación del problema, análisis del problema, diseño e implementación en C, investigación de nuevas funciones como <code>sleep</code>

**Bibliografía.**

Cairo Osvaldo, Metodología de la programación, Ed. Alfaomega 3ª Impresión, México 2005, ISBN: 970-15-0057-01, Págs: 3- 476.

Joyanes Aguilar Luis , Fundamentos de Programación, Ed Mc Graw Hill, 4ª Impresión, España, 2008, ISBN: 8448161114, Págs: 47-73, 74-75, 76-101, 113-141 y 151 -534.

Guardati, Silvia, Estructura de Datos Orientado a Objetos con C ++, Ed.Prentice, México 2007, ISBN: 9702607922, Págs: 1–183.

**NOTA:** Presentar el reporte en un documento (PDF o DOC), el código fuente y la impresión de la pantalla de ejecución.

"Puedes apoyarte de tus compañeros y profesor para aclarar tus dudas"