



Universidad Tecnológica Nacional  
Facultad Regional Resistencia  
Técnico Superior en Programación

# Programación I

## Sentencia SWITCH

# Sentencia SWITCH

## Sentencia Switch

Se puede programar con un grupo de sentencias if then else anidadas, aunque ello puede ser de complicada lectura. Para evitarlo se puede usar la sentencia switch.

Su utilización es:

```
switch (valor)
{

case valor1: <sentencias>
case valor2: <sentencias>
...
default: <sentencias>

}
```

La sentencia switch() en Lenguaje C es una sentencia de selección. Esta sentencia permite seleccionar las acciones a realizar de acuerdo al valor que tome una variable.

En la sentencia switch se comprueba el valor que ingrese el usuario que es el que tomará la variable expresión y lo compara con algunas de las etiquetas que acompañen a cada case. En caso de no encontrar ninguna coincidencia, el valor lo tomará como default.

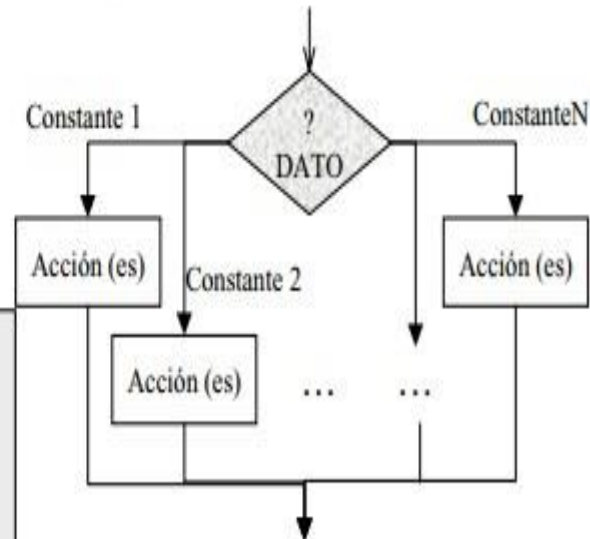
## Sentencia Switch

### Instrucción condicional múltiple : SWITCH

Estructura algorítmica asociada

Formato general de la  
sentencia **switch**

```
switch (selector)
{
    case constante1:
        instrucción1 ó bloque de instrucciones
        break;
    case constante2:
        instrucción2 ó bloque de instrucciones
        break;
    default:
        instrucción2 ó bloque de instrucciones
}
```



Permiten comparar una 'variable' con distintos valores posibles, ejecutando para cada caso una serie de instrucciones específicas.



## Sentencia Switch

### Instrucción condicional múltiple : SWITCH

Formato general de la  
sentencia **switch**

```
switch (selector)
{
    case constante1:
        instrucción1 ó bloque de instrucciones
        break;
    case constante2:
        instrucción2 ó bloque de instrucciones
        break;
    default:
        instrucción3 ó bloque de instrucciones
}
```

El valor de *selector* debe ser un número entero. Puede ser una variable, una expresión ó una llamada a una función.

Cada caso comienza con un **case** y termina con un **break**

¿Qué ocurre si se me olvida algún **break** ?

## **IMPORTANCIA DE BREAK**

**Cuando el control del programa llega a una sentencia switch y se evalúa la expresión de control de la misma, el programa comienza a ejecutar las sentencias presentes en el bloque correspondiente a alguna cláusula case, o quizá a la cláusula default.**

**¿Qué sucede después de ejecutar la última sentencia del bloque? Si el bloque es el último, el programa ejecuta la sentencia inmediatamente siguiente al switch. Pero si no es el último, el programa ejecuta la primera sentencia inmediatamente siguiente al bloque. Y luego la siguiente, y la siguiente, etc. La sentencia break, sirve para hacer que el programa abandone la sentencia switch. Si no se encuentra un break, se ejecuta el siguiente bloque de código, y así hasta hallar un break o salir de la sentencia switch.**

**Dicho de otro modo, las cláusulas case son puntos de entrada al conjunto de líneas de código que hay en el interior del switch. Las sentencias break son puntos de salida, y debemos organizar cuidadosamente ambas cosas para evitar errores que no podrá detectar el compilador.**

# *Lenguaje de Programación C*

---

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int num;
```

```
    printf( "Introduce un número " );
```

```
    scanf( "%i", &num );
```

```
    if ( num==1 ) printf ( "Es un 1\n" );
```

```
    else if ( num==2 ) printf ( "Es un 2\n" );
```

```
        else if ( num==3 ) printf ( "Es un 3\n" );
```

```
            else printf ( "No era ni 1, ni 2, ni 3\n" );
```

```
    getch();
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int num;
```

```
    printf( "Introduce un número " );
```

```
    scanf( "%d", &num );
```

```
    switch( num )
```

```
    {
```

```
        case 1: printf( "Es un 1\n" ); break;
```

```
        case 2: printf( "Es un 2\n" ); break;
```

```
        case 3: printf( "Es un 3\n" ); break;
```

```
        default: printf( "No es ni 1, ni 2, ni 3\n" ) break;
```

```
    }
```

```
    getch();
```

```
    return 0;
```

```
}
```

La sentencia break provoca la salida de switch.

En caso contrario continua la siguiente secuencia case o default aunque no se cumpla la condición.



```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main() /* Escribe el día de la semana */  
{
```

```
    int dia;
```

```
    printf("Introduce el día: ");
```

```
    scanf("%d",&dia);
```

```
    switch(dia)
```

```
    {
```

```
        case 1: printf ("Lunes"); break;
```

```
        case 2: printf ("Martes"); break;
```

```
        case 3: printf ("Miércoles"); break;
```

```
        case 4: printf ("Jueves"); break;
```

```
        case 5: printf ("Viernes"); break;
```

```
        case 6: printf ("Sábado"); break;
```

```
        case 7: printf ("Domingo"); break;
```

```
        default: printf ("No corresponde a ningun dia de la semana\n");break;
```

```
    }
```

```
    return 0;
```

```
    getch();
```

```
}
```



(Zinjal is not just another IDE)

## Resumen:

- La estructura *switch* es especialmente útil cuando la selección se basa en el valor de una variable simple o de una expresión simple denominada expresión de control o selector.
- Los valores de cada *case* del switch han de ser constantes.
- El valor de esta expresión puede ser de tipo *int* o *char*, pero no puede ser del tipo *float* ni *double*.
- La etiqueta *default* marca el bloque de código que se ejecuta por defecto (cuando al evaluar la expresión se obtiene un valor no especificado por los casos anteriores del *switch*).
- La sentencia SWITCH compara solamente igualdad.
- Por cada CASE puede haber 1 o más instrucciones.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
    char car;
    printf("Ingrese un caracter");
    scanf ("%c",&car);

    switch(car)
    {
        case 'a': printf (" Vocal a");break;
        case 'e': printf (" Vocal e");break;
        case 'i': printf (" Vocal i");break;
        case 'o': printf (" Vocal o");break;
        case 'u': printf (" Vocal u");break;
        default: printf("No es vocal");break;
    }
    return 0;
    getch();
}
```



(Zinjal is not just another IDE)

## Ejercicios:

El Estadio “Amigos en el futbol” tiene diversos sectores. El costo de la entrada a los eventos futbolísticos del estadio se asignan en virtud de los sectores del estadio mediante la siguiente tabla:

<i>Sector</i>	<i>Costo de la entrada</i>
Sol general	\$3
Sol preferente	\$5
Sombra	\$8
Tribuna	\$15
Platea	\$20

Se pide construir un programa que permita seleccionar un sector del estadio, ingresar la cantidad de entradas solicitadas y calcular el total a pagar por las entradas.

El programa de contar con un menú de opciones correspondiente con la siguiente imagen:

```
*****
*      BIENVENIDO AL ESTADIO CUSCATLAN      *
*****
* Sectores del estadio                        *
*   A- Sol general                          *
*   B- Sol preferente                      *
*   C- Sombra                             *
*   D- Tribuna                            *
*   E- Platea                             *
*****",
Selecciona la letra del sector del estadio :
```

IMPORTANTE: - Observe que las lecturas de datos están “validadas”, esto significa que en caso que el usuario ingrese valores incorrectos se genera un mensaje que indica al usuario cuál es el error cometido. –

Pruebe lo anterior ingresando un número de sector inexistente por ejemplo 10) o una cantidad de entradas incorrecta (por ejemplo -5).