



**UAEM** | Universidad Autónoma  
del Estado de México  
Centro Universitario UAEM Zumpango  
Ingeniería en Computación



Agosto 2016





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Zumpango  
Ingeniería en Computación

UA: Programación Estructurada  
Tema: Estructuras de control de flujo selectivas

Dr. Asdrúbal López Chau  
alchau@uaemex.mx  
Octubre 2017



# Contenido

1. Propósito de la UA
2. Propósito de la UC 1
3. Estructuras de control de flujo
  - a. Selectivas
4. Ejemplos
5. Conclusiones
6. Referencias



## Propósito de la UA

- Aplicar el paradigma de la programación estructurada para representar en términos de pseudocódigo, la solución de problemas reales automatizables, mostrando en ella el dominio de variables simples, vectores, matrices, registros y modularidad.





## Propósito de la UC 1:

- Identificar las fases de la metodología de programación estructurada para la solución de problemas

## Conocimientos de la UC 1:

- Definir y explicar: Algoritmo, fases de la solución de un programa, sentencia, identificadores, expresiones, reglas de precedencia de signos operacionales, estructuras de datos y estructuras de control.





# Guion explicativo del uso de este material

- En estas diapositivas se explican las estructuras de control de flujo selectivas, aplicables a cualquier lenguaje de programación.
- Se recomienda al docente comenzar con ejemplos simples para que los alumnos entiendan el funcionamiento de cada estructura de control, y paulatinamente avanzar hacia ejemplos más complejos.



# Guion explicativo del uso de este material

- **Se presentan ejemplos completos lenguaje C**, con la finalidad de despertar el interés en los alumnos, y de orientarlos en la implementación de programas simples pero completos en un lenguaje de programación real.
- Se recomienda al docente, usar el lenguaje de programación que considere más apropiado.



## Estructuras de control de flujo

- En los primeros ejemplos de esta UA, se desarrollaron ejemplos en los que se introducen datos y se producen resultados que se imprimen en la pantalla de la computadora.
- En todos estos ejemplos, los programas SON INCAPACES DE TOMAR DECISIONES BASADAS EN LOS DATOS INTRODUCIDOS.



# Estructuras de control de flujo

- En este tema, se aprenderán las principales estructuras de control de flujo selectivas, que permiten hacer programas más interesantes, con capacidad de “tomar decisiones” simples.





# Estructuras de control de flujo

Las estructuras de control de flujo siguientes, son válidas para cualquier lenguaje de programación.

- Selectivas: **IF, IF-ELSE, SWITCH**
- Repetitivas: **FOR, WHILE, DO-WHILE**





# Estructuras de control de flujo

Las estructuras de control de flujo siguientes, son válidas para cualquier lenguaje de programación.

- Selectivas: **IF, IF-ELSE, SWITCH** ← Estas serán revisadas en el presente tema
- Repetitivas: **FOR, WHILE, DO-WHILE** ← Próximo tema



## Estructura de control de flujo IF

Permite ejecutar, o evitar la ejecución de un grupo de sentencias.

La forma general de la estructura IF es la siguiente:

IF condición THEN

    sentencia(s)

END





## Estructura de control de flujo IF

Permite ejecutar, o evitar la ejecución de un grupo de sentencias.

Explicación de **condición**:

**IF** **condición** **THEN**

sentencia(s)

**END**

La **condición** es una sentencia que toma valores **TRUE** o **FALSE**.





# NOTA

En el tema anterior, se explicaron los operadores relacionales. Se recomienda un repaso.

Operador	Se lee
>	Mayor que
<	Menor que
>=	Mayor o igual
<=	Menor o igual que
==	Igual que
!=	Diferente que



## Estructura de control de flujo IF

Permite ejecutar, o evitar la ejecución de un grupo de sentencias.

Funcionamiento de IF:

IF condición THEN

sentencia(s) ← Si condición es TRUE entonces se ejecutan estas sentencias, en caso contrario (condición es FALSE) no se ejecutan.

END





## Estructura de control de flujo IF

**Ejemplo:** Dado un número, imprimir la palabra “Negativo” en caso de que sea menor a cero.

**Solución.**

```
IF numero < 0 THEN  
    Imprime “Negativo”  
END
```





# Estructura de control de flujo IF

**Ejemplo en C:** Dado un número, imprimir la palabra “Negativo” en caso de que sea menor a cero.

```
ejemplo.c x
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero < 0) {
7          printf("Negativo\n");
8      }
9      return 0;
10 }
```



## Estructura de control de flujo IF

**Ejemplo:** Dado un número, imprimir la palabra “Positivo” en caso de que sea mayor a cero.

**Solución.**

```
IF numero > 0 THEN  
    Imprime “Positivo”  
END
```





## Estructura de control de flujo IF

**Ejemplo en C:** Dado un número, imprimir la palabra “Positivo” en caso de que sea mayor a cero.

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero > 0) {
7          printf("Positivo\n");
8      }
9      return 0;
10 }
```



## Estructura de control de flujo IF

**Ejemplo:** Dado un número, imprimir la palabra “Cero” en caso de que sea igual a cero.

**Solución.**

```
IF numero == 0 THEN
    Imprime "Positivo"
END
```



## Estructura de control de flujo IF

**Ejemplo en C:** Dado un número, imprimir la palabra “Cero” en caso de que sea igual a cero.

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero == 0) {
7          printf("Cero\n");
8      }
9      return 0;
10 }
```



# Estructura de control de flujo IF-ELSE

IF-ELSE permite elegir la ejecución de uno de dos bloques de código.

La forma general es la siguiente:

```
IF condición THEN
```

```
    sentencias
```

```
ELSE
```

```
    sentencias
```

```
END
```





## Estructura de control de flujo IF-ELSE

IF-ELSE permite elegir la ejecución de uno de dos bloques de código.

Funcionamiento de IF-ELSE:

IF condición THEN

sentencias

ELSE

sentencias

END

La condición toma valor TRUE o FALSE





## Estructura de control de flujo IF-ELSE

IF-ELSE permite elegir la ejecución de uno de dos bloques de código.

Funcionamiento de IF-ELSE:

IF condición THEN

sentencias ←

Si condición toma valor TRUE  
entonces se ejecutan estas sentencias

ELSE

sentencias ←

Si condición toma valor FALSE  
entonces se ejecutan estas sentencias

END





## Estructura de control de flujo IF-ELSE

**NOTA:** En la condición IF-ELSE, las sentencias son ejecutadas de manera exclusiva.

IF condición THEN

sentencias

ELSE

sentencias

Sólo se ejecutará una de éstas, nunca las dos.

END





## Estructura de control de flujo IF-ELSE

**Ejemplo:** Dado un número, detectar e imprimir si éste es par o impar.

**Solución.**

```
IF numero % 2 ==0 THEN
    Imprime "Par"
ELSE
    Imprime "Impar"
END
```





## Estructura de control de flujo IF-ELSE

**Ejemplo en C:** Dado un número, detectar e imprimir si éste es par o impar.

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero % 2 == 0) {
7          printf("Par\n");
8      } else {
9          printf("Impar\n");
10     }
11     return 0;
12 }
```



## Estructura de control de flujo IF-ELSE

**Ejemplo:** Dado dos números diferentes a cero, indicar si tienen el mismo signo o signos contrarios.

**Solución.**

```
IF numero1*numero2 > 0 THEN
    Imprime "Signos iguales"
ELSE
    Imprime "Signos diferentes"
END
```





## Estructura de control de flujo IF-ELSE

**Ejemplo en C:** Dado dos números diferentes a cero, indicar si tienen el mismo signo o signos contrarios.

```
1  #include<stdio.h>
2  int main() {
3      int numero1, numero2;
4      printf("Introduce número:");
5      scanf("%d",&numero1);
6      printf("Introduce número:");
7      scanf("%d",&numero2);
8
9      if (numero1*numero2 > 0) {
10         printf("Signos iguales\n");
11     } else {
12         printf("Signos diferentes\n");
13     }
14     return 0;
15 }
```





# NOTA

En el tema anterior, se explicaron los operadores lógicos. Se recomienda un repaso de estos operadores.

Operador	Nombre	Produce verdadero si y sólo si
&&	AND	$A \&\& B$ , si A=verdadero y B es verdadero
	OR	$A \    \ B$ , si cualquiera o ambos de A y B son verdaderos
^	XOR	$A \wedge B$ , si A y B tienen valores distintos
!	NOT	$\neg A$ , si A es falso



## Estructura de control de flujo IF-ELSE

**Ejemplo:** Las calificaciones de los alumnos en la UAEM se encuentran en una escala entre 0 y 10. Dada una calificación, identificar si es válida o no. Es decir, verificar si se encuentra o no en el intervalo [0 a 10]

**Solución.**

```
IF score >= 0 && score <= 10 THEN
```

```
    Imprime "Calificación válida"
```

```
ELSE
```

```
    Imprime "Calificación inválida"
```

```
END
```





## Estructura de control de flujo IF-ELSE

**Ejemplo en C:** Las calificaciones de los alumnos en la UAEM se encuentran en una escala entre 0 y 10. Dada una calificación, identificar si es válida o no. Es decir, verificar si se encuentra o no en el intervalo [0 a 10].

ejemplo.c x

```
1  #include<stdio.h>
2  int main() {
3      int score;
4      printf("Introduce calificacion:");
5      scanf("%d",&score);
6      if (score >= 0 && score <= 10) {
7          printf("Calificación válida\n");
8      } else {
9          printf("Calificación inválida\n");
10     }
11     return 0;
12 }
```





# Estructura de control de flujo IF-ELSE

- Pueden anidarse estructuras IF-ELSE, esto es particularmente útil cuando se requiere de probar varias condiciones.

```
IF condición THEN
```

```
    Sentencia
```

```
ELSE IF condición THEN
```

```
    Sentencia
```

```
ELSE
```

```
    Sentencia
```

```
END
```





## Estructura de control de flujo IF-ELSE

- Pueden anidarse estructuras IF-ELSE, esto es particularmente útil cuando se requiere de probar varias condiciones.

```
IF condición THEN
```

```
  Sentencia
```

```
ELSE IF condición THEN
```

```
  Sentencia
```

```
ELSE
```

```
  Sentencia
```

```
END
```

Sólo se ejecutará una de éstas.





## Estructura de control de flujo IF-ELSE

**Ejemplo:** Dado un número, detectar si éste es negativo, positivo o cero, imprimir las palabras “Negativo”, “Positivo” o “Cero” según el caso.

**Solución.**

```
IF numero < 0 THEN
    Imprime "Negativo"
END
IF numero > 0 THEN
    Imprime "Positivo"
END
IF numero == 0 THEN
    Imprime "Cero"
END
```

**OBSERVA QUE SE DEBEN DE PROBAR  
TODAS LAS CONDICIONES.**

**LA SOLUCIÓN CON IF-ELSE SE MUESTRA  
EN LA SIGUIENTE DIAPOSITIVA.**



## Estructura de control de flujo IF-ELSE

**Ejemplo:** Dado un número, detectar si éste es negativo, positivo o cero, imprimir las palabras “Negativo”, “Positivo” o “Cero” según el caso.

**Solución.**

```
IF numero < 0 THEN
    Imprime "Negativo"
ELSE IF numero > 0 THEN
    Imprime "Positivo"
ELSE
    Imprime "Cero"
END
```

Cuando una condición es **TRUE** se ejecuta la sentencia correspondiente, y **YA NO SE CONTINÚAN EVALUANDO** las siguientes condiciones.





## Estructura de control de flujo IF vs IF-ELSE

```
IF numero < 0 THEN
    Imprime "Negativo"
END
IF numero > 0 THEN
    Imprime "Positivo"
END
IF numero == 0 THEN
    Imprime "Cero"
END
```

```
IF numero < 0 THEN
    Imprime "Negativo"
ELSE IF numero > 0 THEN
    Imprime "Positivo"
ELSE
    Imprime "Cero"
END
```



# Estructura de control de flujo IF vs IF-ELSE

ejemplo.c x

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero < 0) {
7          printf("Negativo\n");
8      }
9      if (numero > 0) {
10         printf("Positivo\n");
11     }
12     if (numero == 0) {
13         printf("Cero\n");
14     }
15
16     return 0;
17 }
```

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      if (numero < 0) {
7          printf("Negativo\n");
8      } else if (numero > 0) {
9          printf("Positivo\n");
10     } else {
11         printf("Cero\n");
12     }
13
14     return 0;
15 }
```



# Estructura de control de flujo SWITCH

Esta estructura de control permite seleccionar la ejecución de un bloque de código, dependiendo del valor (o caso) que tenga una variable.

La estructura de SWITCH es la siguiente:

```
SWITCH variable
```

```
CASE c1:
```

```
    sentencias
```

```
CASE c2:
```

```
    sentencias
```

```
DEFAULT
```

```
    sentencias
```





# Estructura de control de flujo SWITCH

Funcionamiento de SWITCH:

SWITCH *variable*

Se busca la coincidencia del valor de la

CASE *c1:*

*variable* con los posibles casos

sentencias

CASE *c2:*

sentencias

DEFAULT

sentencias

END





# Estructura de control de flujo SWITCH

Funcionamiento de SWITCH:

SWITCH *variable*

Se busca la coincidencia del valor de la *variable* con los posibles casos

CASE *c1*:

sentencias

CASE *c2*:

sentencias

Se ejecutan las sentencias de acuerdo a dicha coincidencia

DEFAULT

sentencias

END





# Estructura de control de flujo SWITCH

Funcionamiento de SWITCH:

SWITCH *variable*

CASE c1:

sentencias

CASE c2:

sentencias

DEFAULT



sentencias

END

Si no existe alguna coincidencia, se ejecutan estas sentencias.





# Estructura de control de flujo SWITCH

SWITCH dia

CASE 1:

Imprimir "Lunes"

CASE 2:

Imprimir "Martes"

CASE 3:

Imprimir "Miércoles"

CASE 4:

Imprimir "Jueves"

CASE 5:

Imprimir "Viernes"

CASE 6:

Imprimir "Sábado"

CASE 7:

Imprimir "Domingo"

DEFAULT

Imprimir "No le corresponde algún día"

**Ejemplo:** Dado un número entero entre 1 y 7, imprimir el correspondiente día de la semana que le corresponde.



**Ejemplo en C:** Dado un número entero entre 1 y 7, imprimir el correspondiente día de la semana que le corresponde.

```
1  #include<stdio.h>
2  int main() {
3      int numero;
4      printf("Introduce número:");
5      scanf("%d",&numero);
6      switch(numero) {
7          case 1:
8              printf("Lunes\n");
9              break;
10         case 2:
11             printf("Martes\n");
12             break;
13         case 3:
14             printf("Miércoles\n");
15             break;
16         case 4:
17             printf("Jueves\n");
18             break;
19         case 5:
20             printf("Viernes\n");
21             break;
22         case 6:
23             printf("Sábado\n");
24             break;
25         case 7:
26             printf("Domingo\n");
27             break;
28         default:
29             printf("No le corresponde algún día\n");
30     }
31 }
32
33 return 0;
34 }
```





# NOTAS: Estructura de control de flujo SWITCH

- En general, SWITCH sólo puede usarse con variables enteras. Sin embargo, algunos lenguajes soportan otros tipos como enumeraciones o cadenas.
- No se recomienda usar SWITCH para identificar intervalos.



# Conclusiones

- En este tema se han cubierto las tres estructuras de control de flujo selectivas **IF**, **IF-ELSE** Y **SWITCH**.
- También se han presentado ejemplos de implementaciones completas en lenguaje C. Dependiendo del curso, puede optarse por otro lenguaje de programación.
- Se recomienda ampliamente realizar varios ejercicios tanto en clase como fuera de ella.





## Referencias.

- Nell Dale. Programación y resolución de problemas. Mc Graw Hill, 2010.
- García Bermejo, J. R. Programación Estructurada en C. PRENTICE-HALL, 9788483224236, 2008.
- Joyanes Aguilar, L. Fundamentos de programación. Mc Graw Hill, 2008.
- OSVALDO CAIRO, FUNDAMENTOS DE PROGRAMACION: PIENSA EN C, ISBN 9702608104, Pearson 2006.