

Control de flujo en C

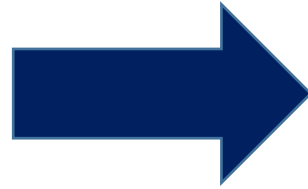
Mg. Ing. Facundo S. Larosa

Informática I

Instituto Universitario Aeronáutico
Universidad de la Defensa Nacional (UNDEF)

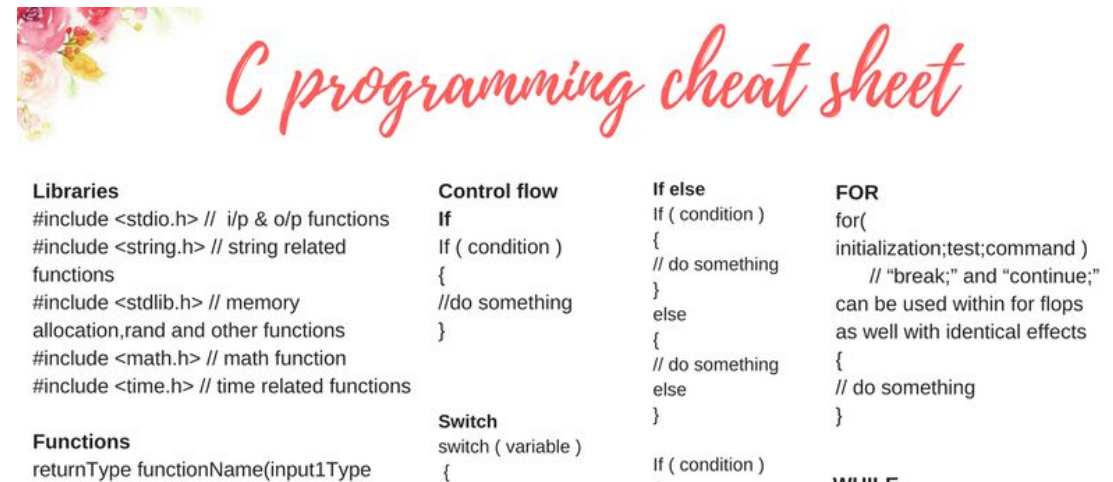
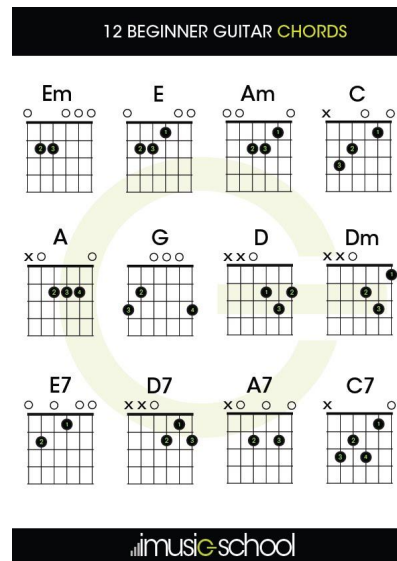
Advertencia I

Como cualquier disciplina compleja, la programación puede ser rutinaria y aburrida al principio... pero sin esos rudimentos introductorios no se puede disfrutar de la fluidez del experto...



Advertencia II

En los comienzos, hasta incorporar la sintaxis del lenguaje es conveniente ayudarse con “hojas de trucos” (*cheatsheets*). Se pueden buscar en internet, de libros, usar esta misma presentación... se pueden imprimir, sacarles una foto con el celular... ¡lo que prefieran!



Control de flujo: Estructuras básicas

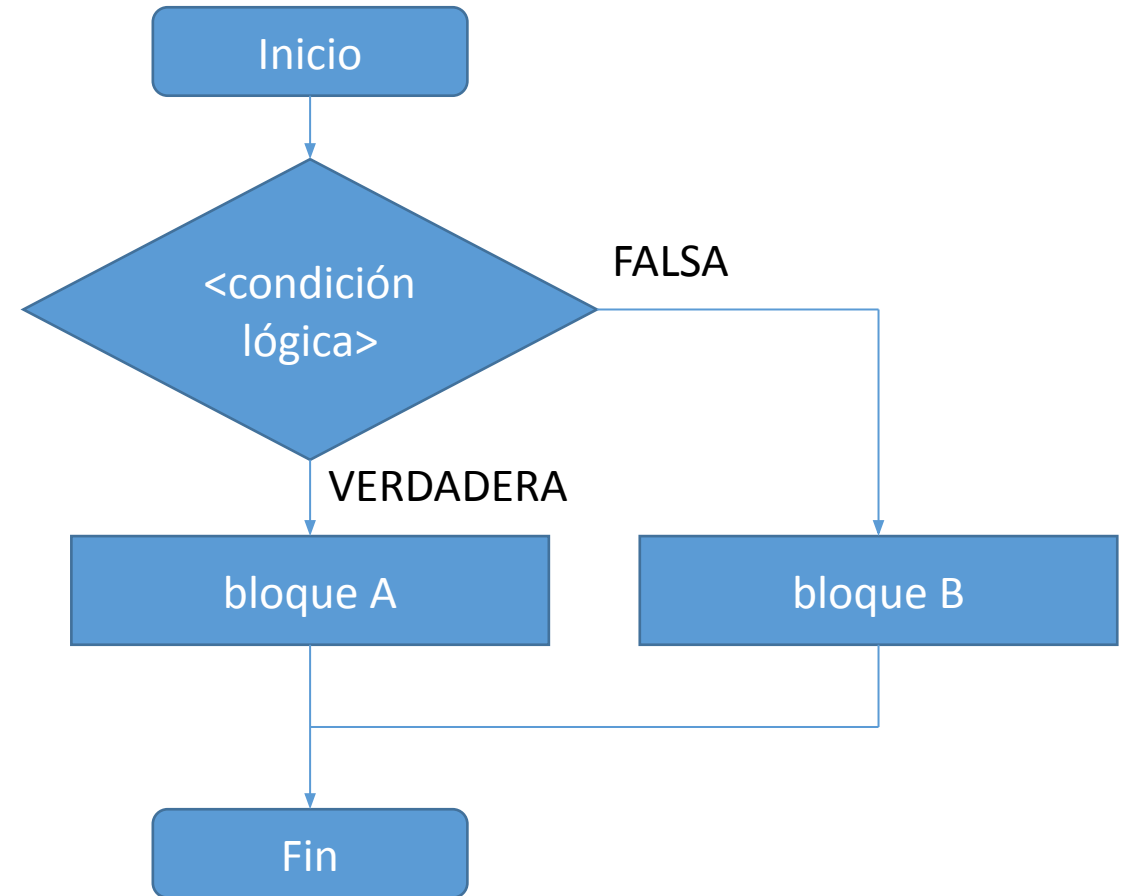
De acuerdo al teorema de la programación estructurada, el lenguaje C ofrece un conjunto de estructuras básicas para la implementación de cualquier tipo de algoritmos:

- **Ejecución condicional:** if, switch
- **Iteración:** while, do while, for

if (salto condicional)

Sintaxis:

```
if (<condición lógica>)  
{  
    bloque A  
}  
else  
{  
    bloque B  
}
```



if (salto condicional)

Ejemplo:

```
/*Cada bloque tiene una sola sentencia*/
```

```
if (nota>=6)
    printf("El alumno está aprobado");
else
    printf("El alumno está desaprobado");
```

if (salto condicional)

Ejemplo:

```
/*Cada bloque tiene varias sentencias*/  
if (nota>=6)  
{  
    printf("El alumno está aprobado");  
    printf("\n\n :D ");  
}  
else  
{  
    printf("El alumno está desaprobado");  
    printf("\n\n : ( ");  
}
```

Operadores relacionales

Se requieren para construir los criterios lógicos

Símbolo	Descripción
==	... igual que ...
!=	... distinto que ...
>	... mayor que ...
>=	... mayor o igual que ...
<	... menor que ...
<=	... menor o igual que ...

A programar (I): ejercicios con “if”

- 1) Realice un programa en el cual se pide al usuario que se ingresen dos números enteros y se imprime un texto indicando si son iguales o no.
Variante: Los números pueden tener parte fraccionaria (reales)
- 1) Realice un programa en el cual se pide al usuario que se ingrese un número entero y se imprima un texto indicando si es positivo, negativo o neutro (cero).
- 1) Realice un programa en el cual se pide al usuario que ingrese un número entero e imprima si es par o impar.
- 1) Realice un programa en el cual el usuario ingresa dos números e indica si el primero es divisible por el segundo o no.

A programar (II): ejercicios con “if”

5) Realice un programa en el cual se pide al usuario que ingrese dos números e imprima cual es el mayor de los dos.

6) Realice un programa en el cual se pide al usuario que ingrese tres números e imprima cual es el mayor de los tres.

7) Realice un programa en el cual se pide al usuario que ingrese los tres coeficientes de una ecuación cuadrática (a , b y c) e indique si sus raíces son: reales y distintas, reales e iguales o complejas.

8) Realice un programa en el cual se pide al usuario que ingrese un carácter e indique si es una letra mayúscula, minúscula o ninguna de las anteriores.

Criterios complejos: Operadores lógicos

Símbolo	Descripción
	O (OR)
&&	Y (AND)
!	NO (NOT)

Tablas de verdad

A	B	A B
FALSA	FALSA	FALSA
FALSA	VERDADERA	VERDADERA
VERDADERA	FALSA	VERDADERA
VERDADERA	VERDADERA	VERDADERA

A	B	A && B
FALSA	FALSA	FALSA
FALSA	VERDADERA	FALSA
VERDADERA	FALSA	FALSA
VERDADERA	VERDADERA	VERDADERA

A	! A
FALSA	VERDADERA
VERDADERA	FALSA

Construcción de condiciones lógicas I

Existe una particularidad en C, una condición lógica formada únicamente por una variable o constante es:

- **Falsa:** Si la variable es cero
- **Verdadera:** Si la variable es distinta de cero

```
/*¿Qué imprime este ejemplo*/  
int var=10;  
if(var)  
{  
    printf("\n var=%d",var) ;  
}
```

Construcción de condiciones lógicas II

Los operadores relacionales (`==`, `!=`, `>=`, `>`, `<` y `<=`) se resuelven antes que los operadores lógicos (`&&`, `||`, `!`). ¿Qué decisiones toman los *ifs* siguientes?

```
A=-2; /*Para todos los ejemplos*/
```

```
B=3;
```

```
C=5;
```

```
if (! (A+2) && C>B)
```

```
/*VERDADERA*/
```

```
else
```

```
/*FALSA*/
```

```
if (A)
```

```
/*VERDADERA*/
```

```
else
```

```
/*FALSA*/
```

```
if (A>0 && B==3 || C > 3)
```

```
/*VERDADERA*/
```

```
else
```

```
/*FALSA*/
```

```
if (B-A != C)
```

```
/*VERDADERA*/
```

```
else
```

```
/*FALSA*/
```

A programar (III): ejercicios con “if”

9) Realice un programa en el cual se pide al usuario que ingrese las longitudes de los lados de un triángulo e indique si el triángulo es equilátero, isósceles o escaleno.

10) Realice un programa en el cual se pide al usuario que ingrese una letra minúscula e indique si es una vocal o una consonante. Si no es una letra minúscula, el programa termina indicando un error.

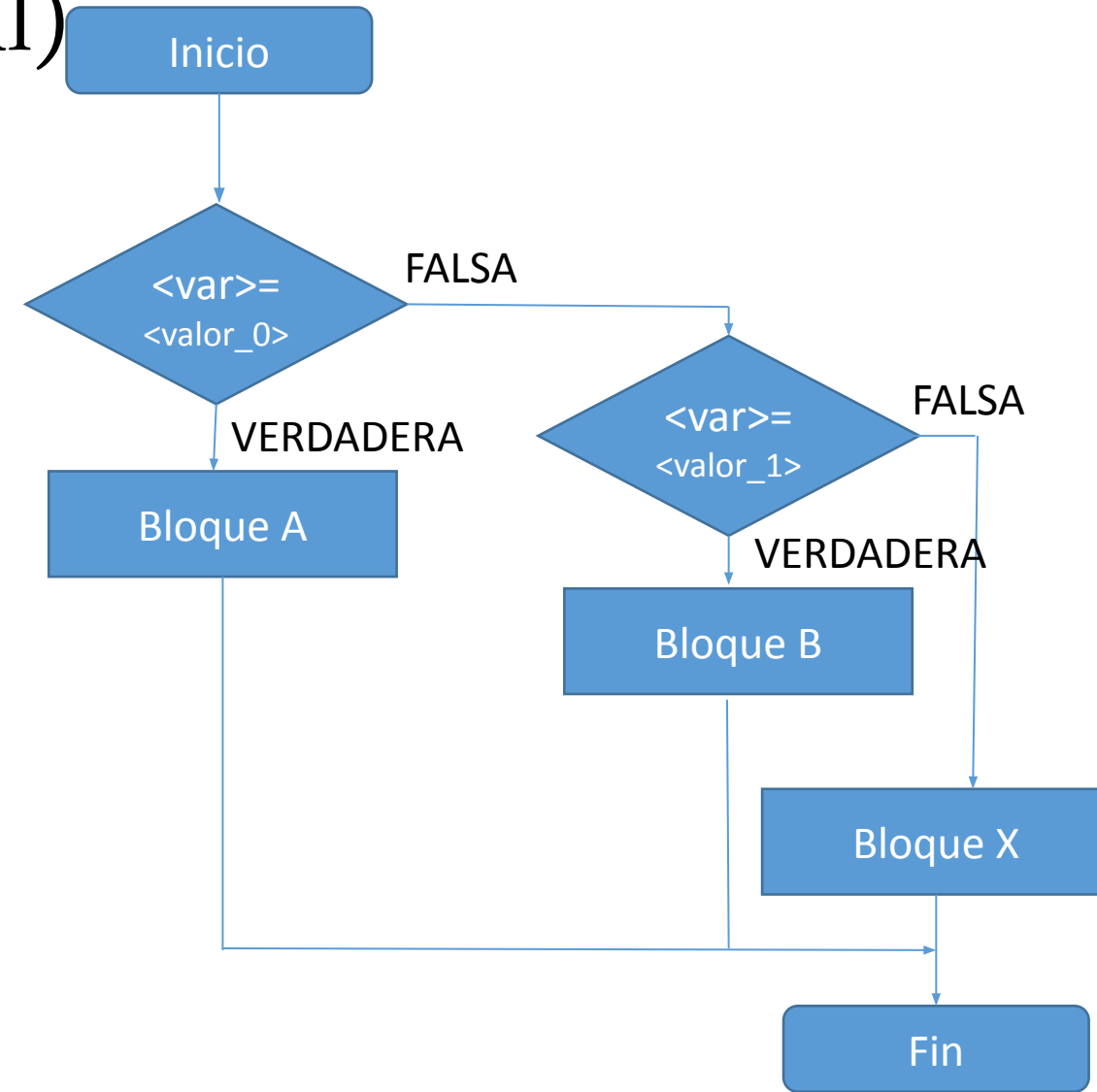
11) Realice un programa en el cual se pide al usuario que ingrese tres números en cualquier orden y luego los imprima ordenados de mayor a menor.

12) Realice un programa en el cual se pide al usuario que ingrese un par de coordenadas (X,Y) correspondientes a un punto del plano y el programa indique en que cuadrante se encuentra.

switch case (salto condicional)

Sintaxis:

```
switch (<variable entera>)  
{  
    case <valor_0>: <bloque A>;  
        break;  
    case <valor_1>: <bloque B>;  
        break;  
    /*Se pueden poner más "case"*/  
    default: <bloque X>;  
        break;  
}
```



switch case (salto condicional)

Ejemplo: Programa para calcular áreas de diferentes figuras geométricas

```
int sel;  
float a,b;  
  
printf("Ingrese la figura cuya área desea calcular:  
      \n0.  Círculo  
      \n1.  Cuadrado  
      \n2.  Rectángulo");  
  
scanf("%d", &sel);
```

Continúa ejemplo anterior:

```
switch (sel)
{
    case 0: printf("\n\nIngrese radio del círculo: ");
            scanf("%f",&a);
            printf("El área del círculo es: %f", 3.1416*a*a);
            break;
    case 1: printf("\n\nIngrese lado del cuadrado: ");
            scanf("%f",&a);
            printf("El área del cuadrado es: %f", a*a);
            break;
    case 2: printf("\n\nIngrese lados del rectángulo: ");
            scanf("%f %f",&a,&b);
            printf("El área del rectángulo es: %f", a*b);
            break;
    default: printf("\nLa opción ingresada es incorrecta");
            break;
}
```

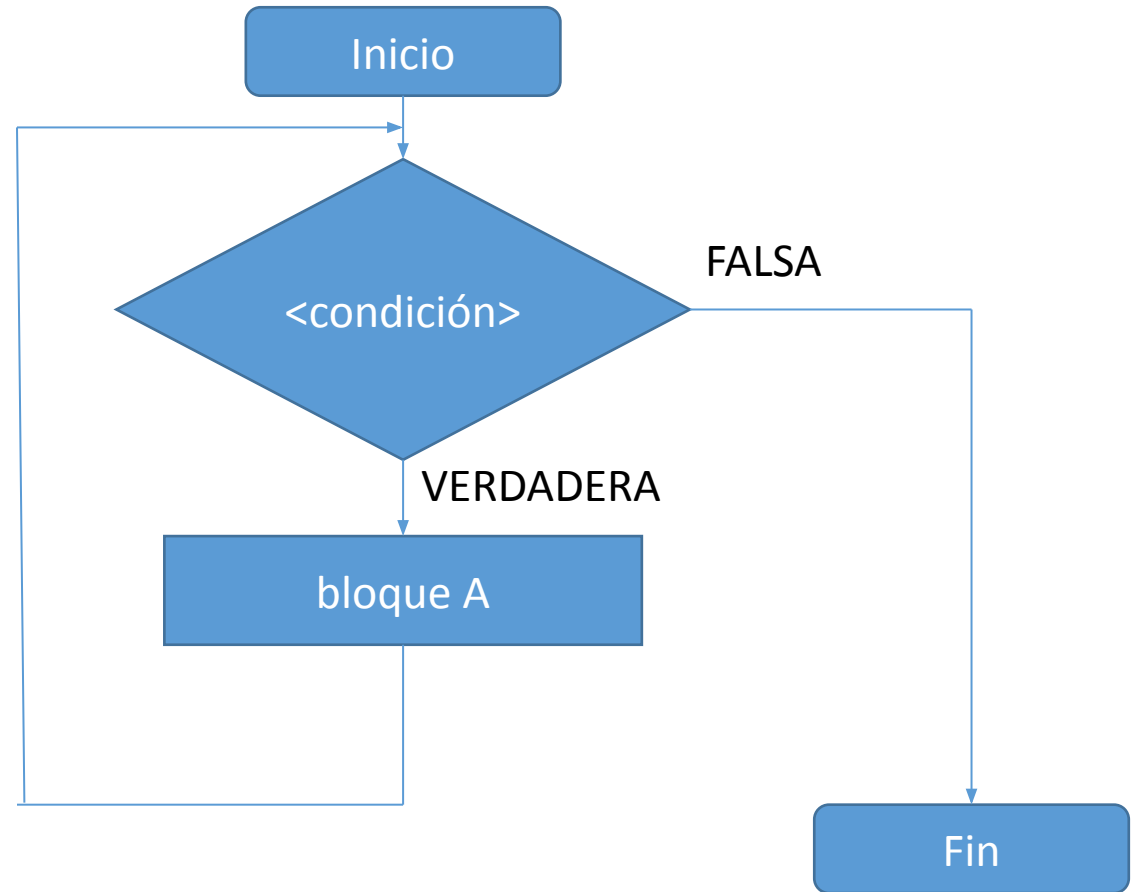
A programar (IV): ejercicios con “switch”

- 1) Realice un programa en el cual se ingrese el número de mes y el programa imprima el nombre del mes y la cantidad de días que tiene.
- 2) Realice un programa que de al usuario la posibilidad de realizar cuatro operaciones (suma, resta, multiplicación y división) de acuerdo a una opción numérica y luego solicite los operandos y dé el resultado.
- 3) Realice un programa en el cual se ingresa una nota del 1 al 10 y se imprime una leyenda indicando una valoración de la nota: sobresaliente (10), distinguido (9), bueno (7 u 8), aprobado (6), desaprobado (5 o menos).

while (lazo condicional)

Sintaxis:

```
while (<condición>)  
{  
    bloque A  
}
```



while (lazo condicional)

Ejemplo:

```
int num=0;
```

```
while (num<10)
```

```
{
```

```
    printf("\n num=%d", num) ;
```

```
    num++;
```

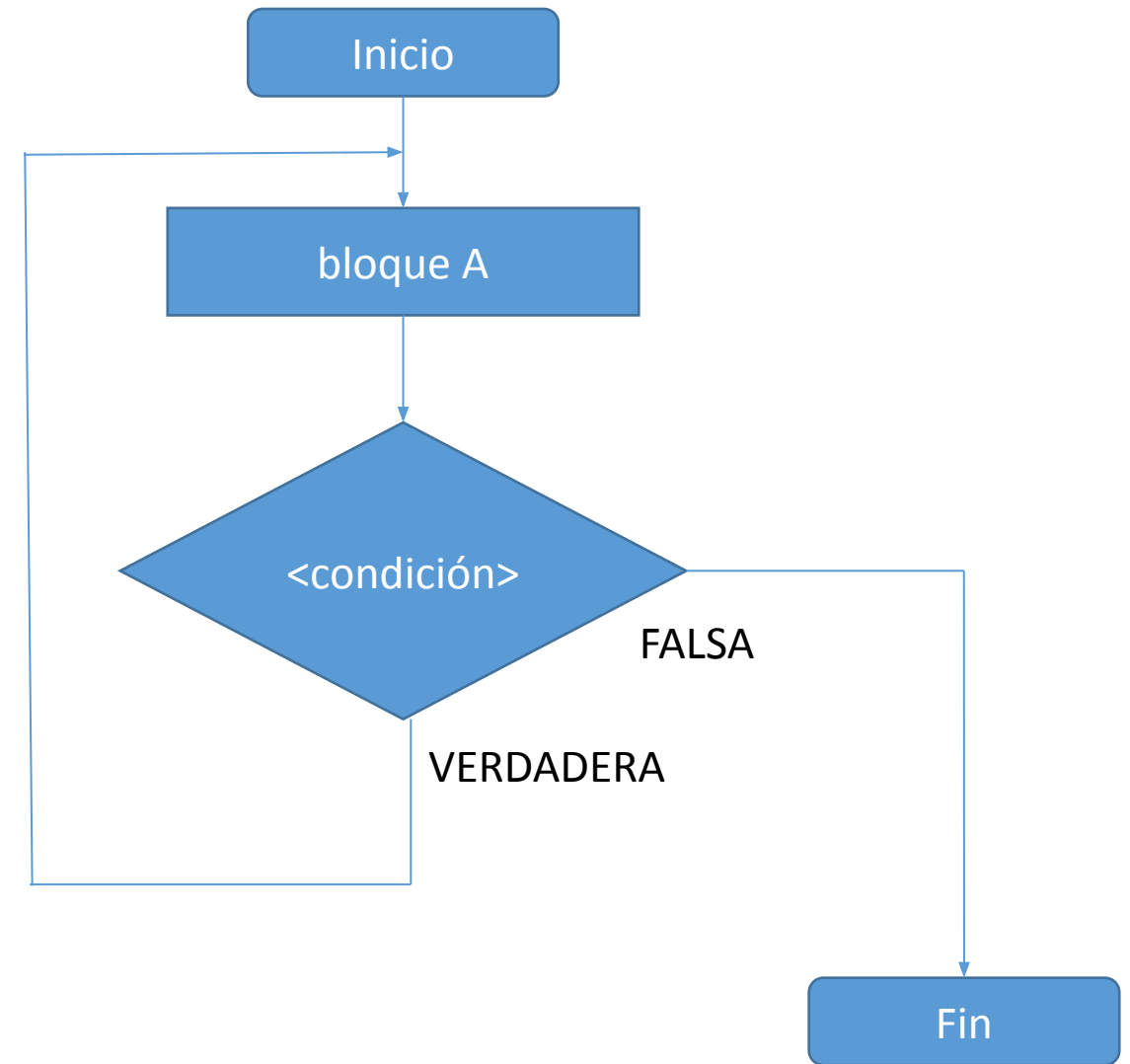
```
}
```

```
/*¿Qué imprime este ejemplo... Hagamos una prueba de  
escritorio...*/
```


do while (lazo condicional)

Sintaxis:

```
do  
{  
    bloque A  
}while (<condición>);
```



do while (lazo condicional)

Ejemplo:

```
int num=20;
```

```
do  
{  
    printf("\n num=%d", num) ;  
    num=num/2;  
}while (num>0) ;
```

```
/*¿Qué imprime este ejemplo... Hagamos una prueba de  
escritorio...*/
```

A programar (V): ejercicios con “while” y “do while”

- 1) Realice un programa donde se le pide al usuario que ingrese un número del 1 al 9 y se imprime la tabla de multiplicar (múltiplos de 1 a 9) de ese número. Si el número ingresado no está en el rango pautado se sale del programa con un mensaje de error.
- 1) Realice un programa donde se le pide al usuario que ingrese números reales, los cuales se van “acumulando” (sumando) hasta que el usuario ingresa el número cero. Finalmente, se muestra el total al usuario.
- 1) Realice un programa donde se le pide al usuario que ingrese dos letras cualquiera y el programa imprima todas las letras comprendidas entre ellas. Asuma que la primer letra es menor que la primera y que el usuario no se equivoca (por ejemplo, que no ingresa caracteres que no son letras).

Variante: Agregue comprobaciones para evitar que el usuario ingrese valores que no correspondan a letras y que no ingrese una letra “mayor” en primer lugar (por ejemplo, ‘z’ y ‘h’).

A programar (VI): ejercicios con “while” y “do while”

4) Realice un programa donde se le pide al usuario que ingrese dos números enteros (base y exponente) e imprima la potencia. Asuma que ambos números son positivos. Variante: Mejore el programa anterior para que pueda procesar bases y exponentes negativos. Contemple el caso en el cual la base es cero y el exponente es negativo indicando una situación de error al usuario.

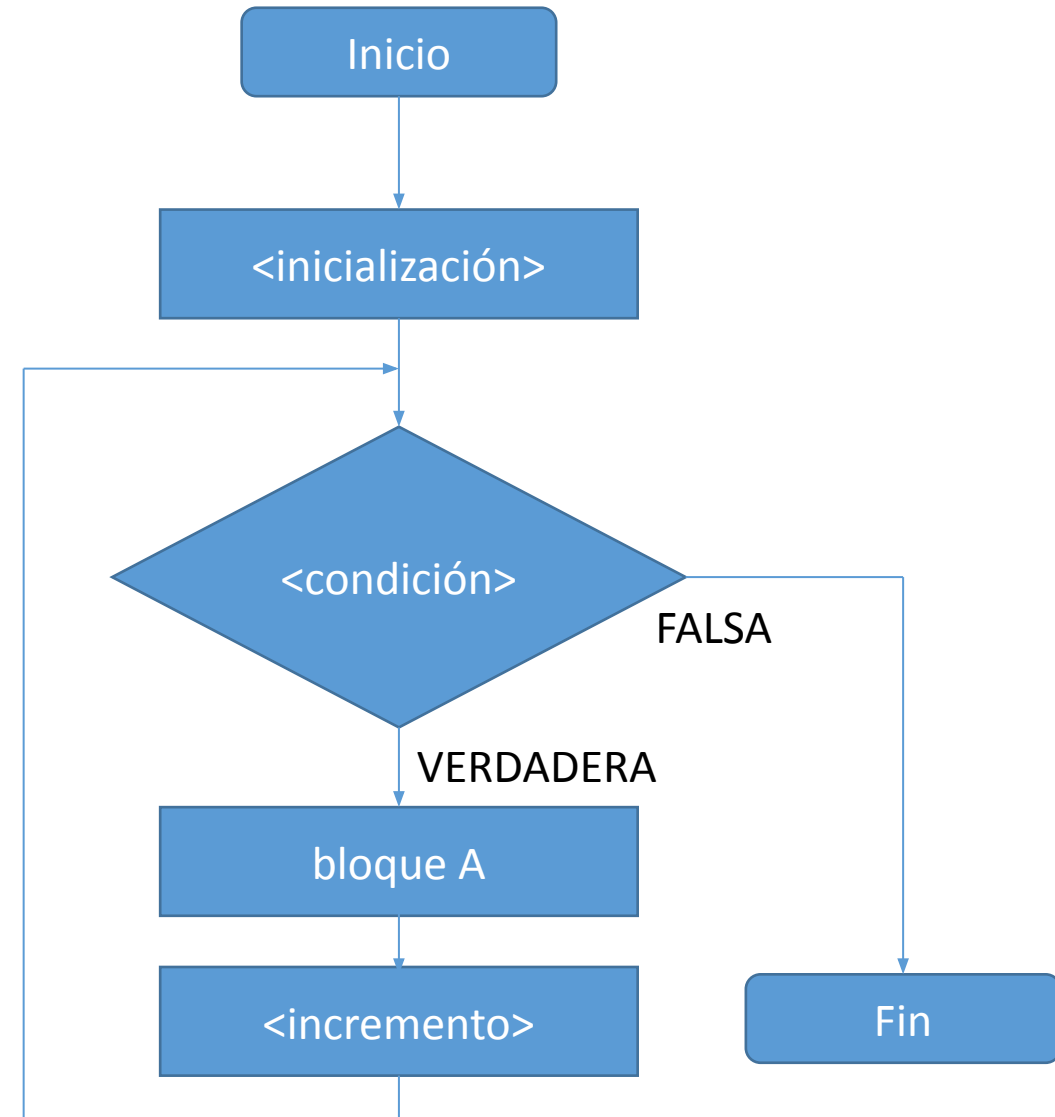
5) Realice un programa donde se le pide al usuario que ingrese un número por teclado. El programa indica si el número es primo o no [los números primos son aquellos cuyos únicos divisores son la unidad y el mismo número].

6) Realice un programa donde se le pide al usuario que ingrese un número entero y calcule su factorial

for (lazo condicional)

Sintaxis:

```
for (<inic>;<cond>;<incr>)  
{  
    bloque A  
}
```



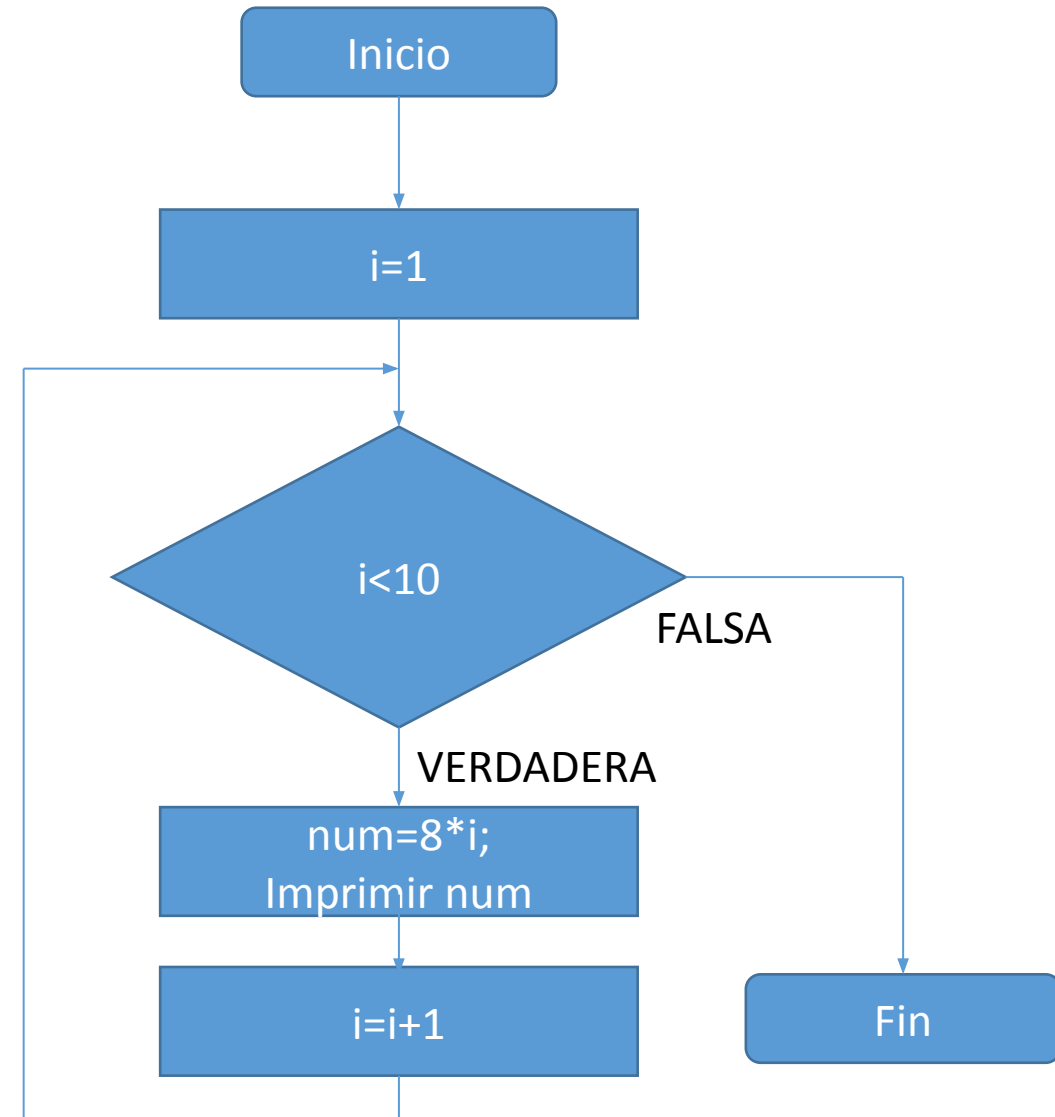
for (lazo condicional)

Ejemplo:

```
int i;
```

```
for (i=1;i<10;i++)  
{  
    num=8*i;  
    printf("\n num=%d",num) ;  
}
```

/¿Qué imprime este ejemplo*/*



A programar (VII): ejercicios con “for”

- 1) Implemente los ejercicios que no haya realizado con “while” o “do while” utilizando la sentencia “for”. Elija aquellos que se adapten mejor a dicha implementación.
- 2) 2) Se le pide al usuario que ingrese un número por teclado. El programa indica si el número es primo o no [los números primos son aquellos cuyos únicos divisores son la unidad y el mismo número].

Apéndice: Sentencia *break*

La sentencia *break* permite salir del lazo de control que me encuentro ejecutando.

```
/*¿Qué imprime este ejemplo*/  
for(i=0;i<10;i++)  
{  
    if (i==5) break;  
  
    printf("\ni vale %d",i) ;  
}
```

Apéndice: Sentencia *continue*

La sentencia ***continue*** permite saltar las sentencias que siguen en el bloque e ir directamente a la próxima evaluación de la condición

```
/*¿Qué imprime este ejemplo*/  
for (i=0; i<10; i++)  
{  
    if (i==5) continue;  
  
    printf("\ni vale %d", i);  
}
```

Ejercicio integrador

Se desea hacer un programa que opere como una calculadora simple. El programa pide en un menú las siguientes opciones:

0. Suma
1. Resta
2. Multiplicación
3. División
4. Potenciación [el exponente es un número entero positivo o negativo]
5. Salir

Una vez seleccionada la opción, el programa solicita el ingreso de los datos y muestra el resultado por pantalla. Luego, borra la pantalla y vuelve a imprimir el menú hasta que el usuario decida salir. [Nota: para borrar la pantalla hacer `system("clear")`, `system` es una función de la biblioteca "`stdlib.h`"]. Todas las operaciones usan dos operandos.

Bibliografía

- Schildt, H. , “C Manual de referencia”, Capítulo 3
- Deitel, “Cómo programar en C/C++”, Capítulo 3
- Gottfried, B. , “Programación en C”, Capítulo 6
- Argibay J. , “C para Ingeniería Electrónica”, Capítulo 3
- Ceballos, F. , “Enciclopedia del lenguaje C”, Capítulo 4
- Kernighan B, Ritchie D. , “El lenguaje de programación C”,
Capítulo 3