



Codificación en C de un algoritmo en Diagrama de Flujo

Introducción al lenguaje C

Estructura general de un programa en C

```
//*****  
/*Qué hace el programa. Enunciado. Comentarios  
Fecha de creación y nombre del autor  
*/  
//*****  
//Sección del preprocesador  
#include <nombre_del_archivo.h> // Inclusión de librerías  
#include <nombre_del_archivo.h>  
  
//Sección de declaraciones globales  
  


Lugar para declaración de variables, constantes y prototipos de funciones

  
//Sección función principal  
int main()  
{// Empieza el bloque de la función principal  
  


Definición de variables y constantes locales

  


Sentencias del programa

  
return 0;  
}// Final del bloque de la función principal  
  
//Definición de funciones
```

Creación de un programa:

- Escribir el programa usando un editor de texto. Este archivo constituye el código fuente del programa.
- Compilar el código fuente → código objeto.
- Enlazar o linkear el código objeto con la librerías correspondientes → código ejecutable.

El programa obtenido recibe datos y nos entrega resultados luego de la ejecución.

Los tipos fundamentales de datos en C son:

- Enteros: int
- Reales: float.
- Caracteres (letras, dígitos, símbolos y signos de puntuación): char



Operadores

Según el número de operadores sobre los que actúa un operador, estos se clasifican en: unitarios, binarios y ternarios:

Operadores aritméticos	+	suma
	-	resta
	*	multiplicación
	/	división
	%	módulo
Operadores relacionales	==	igual que
	!=	distinto que
	<	menor que
	>	mayor que
	<=	menor o igual que
	>=	mayor o igual que
Operadores lógicos	&&	AND
	!	NOT
		OR

Estructuras básicas de la programación estructurada

- Secuencial
- Selección

If

```
if (expresión lógica)
    sentencia;
```

If – else

```
if (expresión)
{
    sentencia;
}
else
{
    sentencia;
}
```

Cuando el cuerpo del if contiene más de una sentencia, es decir un bloque de código, este debe ir encerrado entre llaves.

- Repetición

While

```
while (expresión lógica)
{
    sentencia;
}
```



Do-while

```
do {  
    sentencia;  
}  
while (expresión);
```

Funciones de entrada/salida de datos:

Para tomar datos, generalmente por teclado usamos la función scanf. La sintaxis es:

```
scanf(cadena_formato, &nombre_variable)
```

Para mostrar datos, generalmente por pantalla usamos printf:

```
printf(cadena_formato, nombre_variable)
```

La cadena de formato hace referencia al tipo de dato que queremos mostrar o leer. En el caso de que la variable x se haya definido de tipo entero colocamos para leer el valor que tomará la variable:

```
scanf("%d", &x)
```

Para mostrar el valor de x colocamos:

```
printf("%d", x)
```

Ejemplo: Dado un número entero N presentar por pantalla los números de 1 a N.

```
# include <stdio.h>  
int main()  
{  
    int N,i;  
    printf("\n\tIngrese la cantidad de números a mostrar: ");  
    scanf("%d", &N);  
    i=1;  
    while(i<=N)  
    {  
        printf("\n\t\t\t %d",i);  
        i = i+1;  
    }  
    return(0) ;  
}
```