

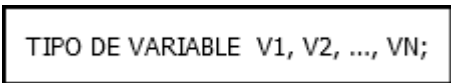
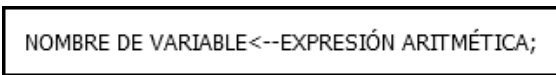
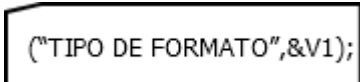
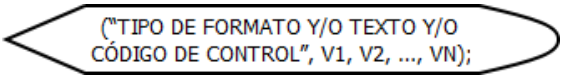


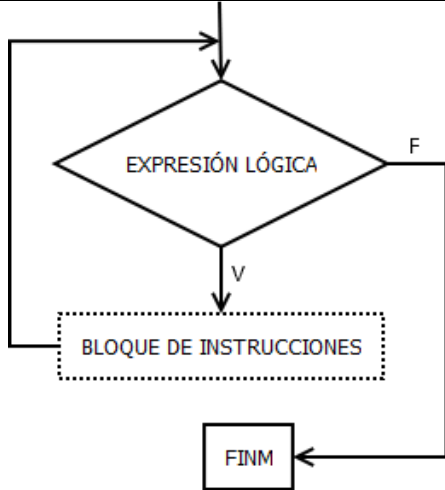
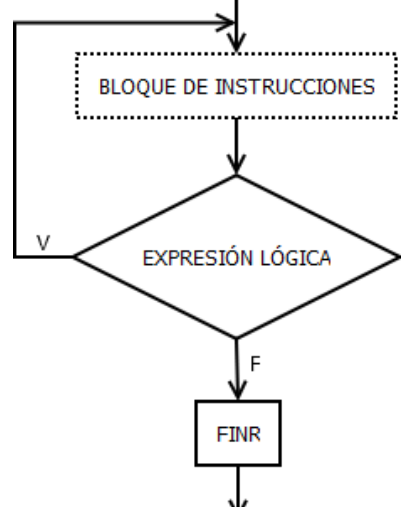
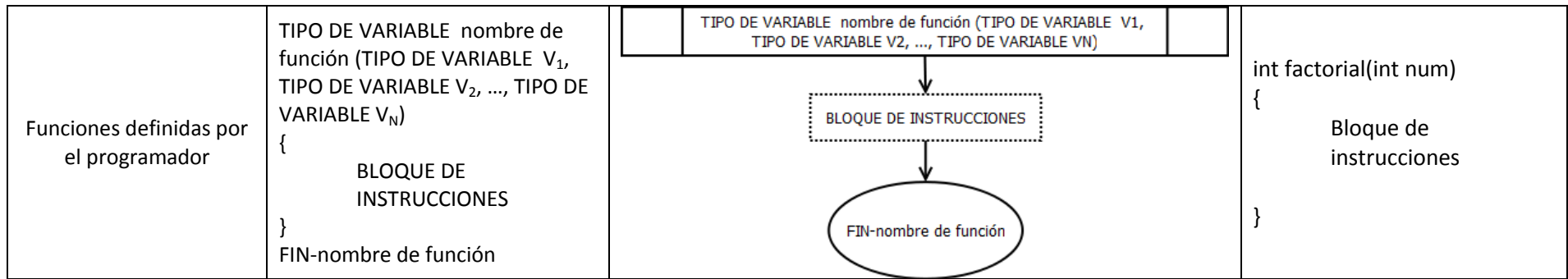


Uso	Pseudocódigo	Diagrama de flujo	Codificación
Inicio	INICIO		<pre>#include <conio.h> #include <stdio.h> int main(void) {</pre>
Fin	FIN		<pre>}</pre>
Declaración de variables	TIPO DE VARIABLE V_1, V_2, \dots, V_N ;		<pre>int v1,v2,...,vN; float D,E; char f,G, E[5];</pre>
Asignación	NOMBRE DE VARIABLE ← EXPRESIÓN ARITMÉTICA;		<pre>suma = var * 5;</pre>
Lectura de datos	LEER("TIPO DE FORMATO", & V_1);		<pre>scanf("%f", &a1); scanf("%d,%f",&a,&b);</pre>
Despliegue de datos	DESPLEGAR("TIPO DE FORMATO Y/O TEXTO Y/O CÓDIGO DE CONTROL", V_1, V_2, \dots, V_N);		<pre>printf("%d",x); printf("El resultado es: %i",x);</pre>
*Conector dentro de la página	-		-
*Conector fuera de la página	-		-

<p>De acuerdo a una condición, ejecución de un grupo u otro de sentencias</p>	<pre> SI(EXPRESIÓN LÓGICA)ENTONCES { BLOQUE DE INSTRUCCIONES CASO VERDADERO } CASO CONTRARIO { BLOQUE DE INSTRUCCIONES CASO FALSO } FINSI </pre>		<pre> if(expresión lógica) { Bloque de instrucciones /*caso verdadero*/ } else { Bloque de instrucciones /*caso falso*/ } </pre>
<p>Ejecución de un grupo de sentencias si se cumple una condición</p>	<pre> SI(EXPRESIÓN LÓGICA) ENTONCES { BLOQUE DE INSTRUCCIONES CASO VERDADERO } FINSI </pre>		<pre> if(expresión lógica) { Bloque de instrucciones /*caso verdadero*/ } </pre>

<p>Ejecución de un grupo de sentencias mientras se cumpla una condición</p>	<pre> MIENTRAS(EXPRESIÓN LÓGICA) { BLOQUE DE INSTRUCCIONES CASO VERDADERO } FINM </pre>		<pre> while(expresión lógica) { Bloque de instrucciones /*caso verdadero*/ } </pre>
<p>Ejecución de un grupo de sentencias hasta que se cumpla una condición</p>	<pre> REALIZA { BLOQUE DE INSTRUCCIONES } MIENTRAS(EXPRESIÓN LÓGICA); </pre>		<pre> do { Bloque de instrucciones /*caso verdadero después de evaluar la exp. lóg.*/ }while(expresión lógica); </pre>

<p>De acuerdo al valor de una variable, ejecución de un grupo u otro de sentencias</p>	<pre> SELECCIONAR (VARIABLE) { CASO VARIABLE=VALOR1: BLOQUE DE INSTRUCCIONES 1 CASO VARIABLE=VALOR2: BLOQUE DE INSTRUCCIONES 2 ⋮ CASO VARIABLE=VALORN: BLOQUE DE INSTRUCCIONES N CASO VARIABLE= VALOR DIFERENTE: BLOQUE DE INSTRUCCIONES DE CASO DIFERENTE } FINS </pre>		<pre> switch(variable) { case 'a': printf("Se ha pulsado una a."); break; case 'e': printf("Se ha pulsado una e."); break; case 'i': printf("Se ha pulsado una i."); break; default: printf("Error"); break; } </pre>
<p>Ejecución de un grupo de sentencias un número determinado de veces</p>	<pre> PARA (EXPRESIÓN1; EXPRESIÓN2; EXPRESIÓN3) { BLOQUE DE INSTRUCCIONES } FINP </pre>		<pre> for(i=0; i<=n; i++) { suma=suma+i; } </pre>



*Símbolos únicamente utilizados en diagramas de flujo.

Reglas para la elaboración de diagramas de flujo

1. Las instrucciones se escriben y se ejecutan de arriba hacia abajo y de izquierda a derecha.
2. Los símbolos se unen con líneas de flujo, las cuales tienen en la punta una flecha que identifica la dirección en que fluye la información.
3. Todo símbolo (excepto las líneas de flujo) llevará en su interior información que indique su función exacta y unívoca, evitando el uso de muchas palabras.
4. La mayor parte de los símbolos pueden tener más de una línea de entrada.
5. Todos los símbolos del diagrama **no** pueden tener más de una salida si no son símbolos de decisión.
6. Las líneas de flujo no pueden cruzarse.
7. Se deben utilizar solamente líneas de flujo rectas horizontales y/o verticales, nunca diagonales.
8. Para evitar el cruce de líneas se pueden utilizar conectores, pero se limita su uso para casos en los que no haya más opciones.
9. No deben quedar líneas sin conectarse.

Tipo de variable	Codificación en lenguaje C
Entero	int
Entero largo	long int
Real	float
Real largo	long float, double
Carácter corto	char
Carácter largo	char

Tipo de formato	Codificación en lenguaje C
Entero	%d, %i
Entero largo	%ld, %li
Real	%f
Real largo	%lf
Carácter	%c
Cadena de caracteres	%s

Código de control	Codificación en lenguaje C
Salto de línea	\n
Tabulador	\t