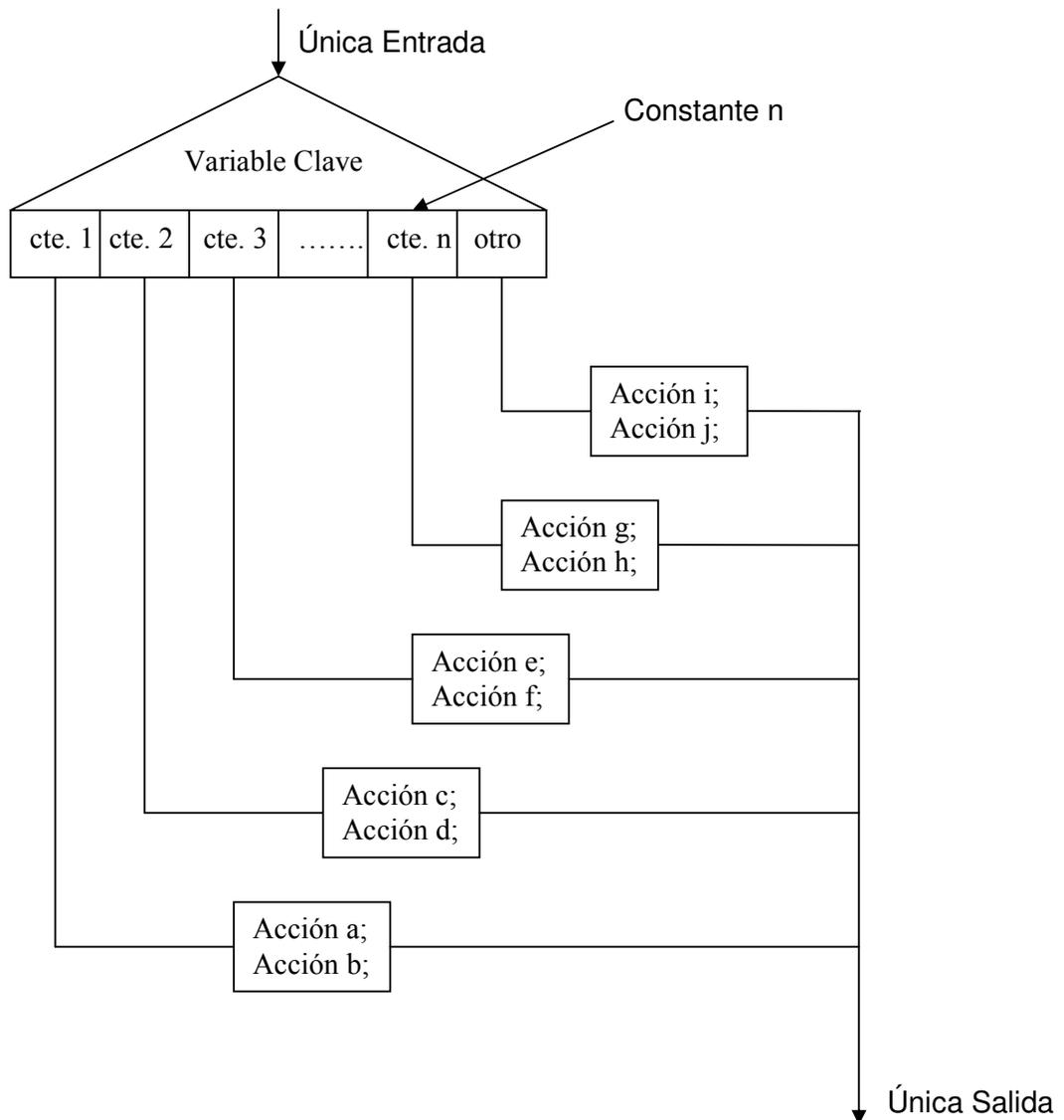


Estructura Selectiva Múltiple

La estructura que se va a describir a continuación desarrolla una función similar a la de la selectiva doble con múltiples ramificaciones, aunque como se puede ver presenta también importantes diferencias. La forma general de la estructura selectiva Múltiple o codificada en C “*switch*” tiene el siguiente en diagrama de flujo:

**Diagrama de Flujo de la estructura selectiva múltiple
“Según”o “switch”**



Explicación:

Se evalúa la **variable clave** y si coincide con el valor constante **cte1**, se ejecuta **acción a** seguida de **acción b** y salta al final de la estructura.

Si no, se evalúa la **variable clave** y si coincide con el valor constante **cte2**, se ejecuta **acción c** seguida de **acción d** y salta al final de la estructura.

Si no, se evalúa la **variable clave** y si coincide con el valor constante **cte3**, se ejecuta **acción e** seguida de **acción f** y salta al final de la estructura.

Si no, se evalúa la **variable clave** y si coincide con el valor constante **cten**, se ejecuta **acción g** seguida de **acción h** y salta al final de la estructura.

Si no, finalmente se evalúa la **variable clave** y si no coincidió con el valor constante **cte1**, ni con **cte2**, **cte3**, **cten**, se ejecuta **acción i** seguida de **acción j** y finaliza.

Como se ve en el pseudocódigo al terminar el bloque de sentencias en cada caso encontramos la palabra salto, el efecto de la misma, es dar por terminada la ejecución de la estructura **Según**.

Pseudocódigo

```

Según (variable clave)
{ caso cte1: Acción i;      /* estas acciones se ejecutan cuando
    Acción j;      variable clave == cte1 da verdadero*/
    salto;
  caso cte2: Acción e;      /* estas acciones se ejecutan cuando
    Acción f;      variable clave== cte2 da verdadero */
    salto;
  -----
  caso cten: Acción c;      /* estas acciones se ejecutan cuando
    Acción d;      variable clave == cten da verdadero*/
    salto;
  otro: Acción a;          /* estos finalmente si variable clave != cte1&&
    Acción b;          variable clave!= cte2&& variable clave != cten
    }                  da verdadero*/

```

Codificación en C

```

switch (expresion)
{
    case expresion_cte_1:
        sentencia_1;
    case expresion_cte_2:
        sentencia_2;
    ...
    case expresion_cte_n:
        sentencia_n;
    [default:
        sentencia;]
}

```

Explicación:

Se evalúa **expresión** y se considera el resultado de dicha evaluación. Si dicho resultado coincide con el valor constante **expresion_cte_1**, se ejecuta **sentencia_1** seguida de **sentencia_2**, **sentencia_3**, ..., **sentencia**.

Si el resultado coincide con el valor constante **expresion_cte_2**, se ejecuta **sentencia_2** seguida de **sentencia_3**, ..., **sentencia**.

En general, se ejecutan todas aquellas sentencias que están a continuación de la **expresion_cte** cuyo valor coincide con el resultado calculado al principio.

Si ninguna **expresion_cte** coincide se ejecuta la **sentencia** que está a continuación de **default**.

Si se desea ejecutar únicamente una **sentencia_i** (y no todo un conjunto de ellas), basta poner una sentencia **break** a continuación. El efecto de la sentencia **break** es dar por terminada la ejecución de la sentencia **switch**.

Existe también la posibilidad de ejecutar la misma **sentencia_i** para varios valores del resultado de **expresion**, poniendo varios **case expresion_cte** seguidos. El siguiente ejemplo ilustra las posibilidades citadas:

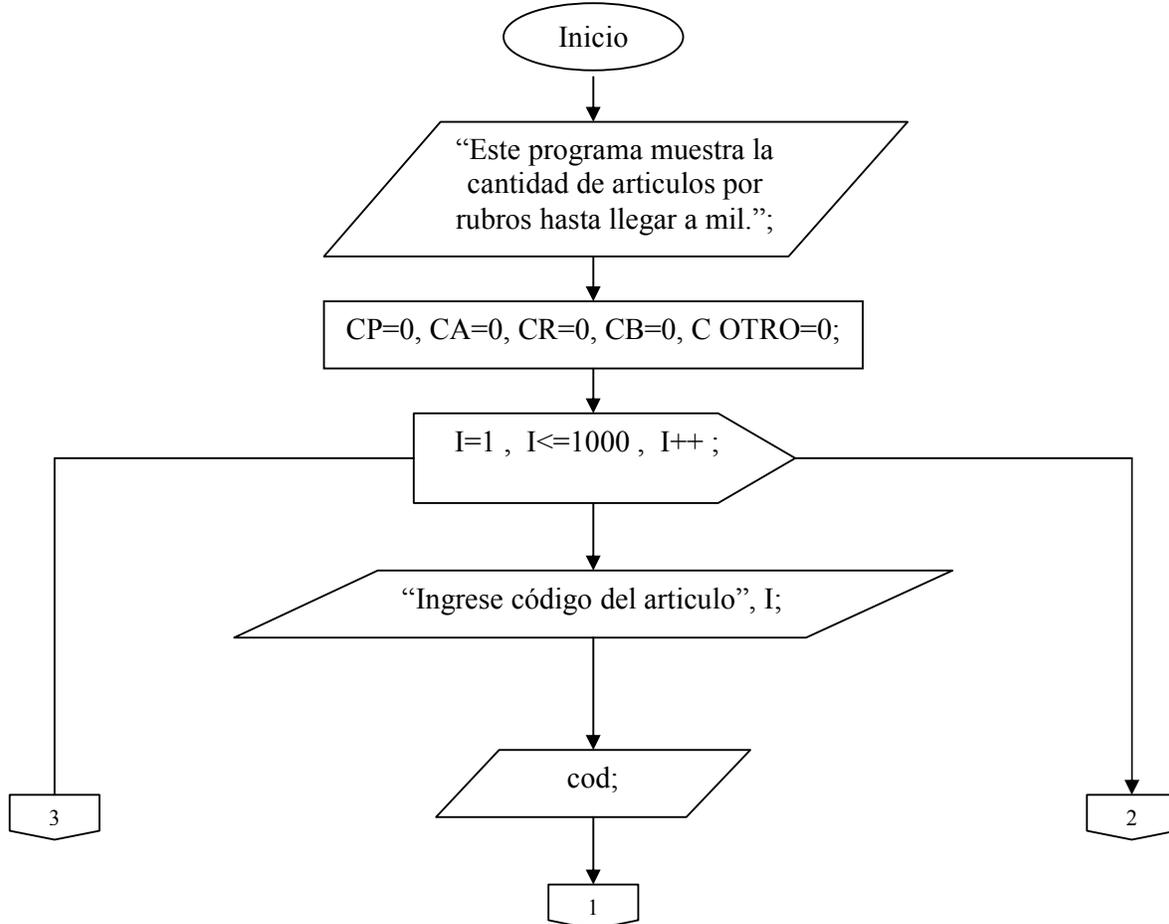
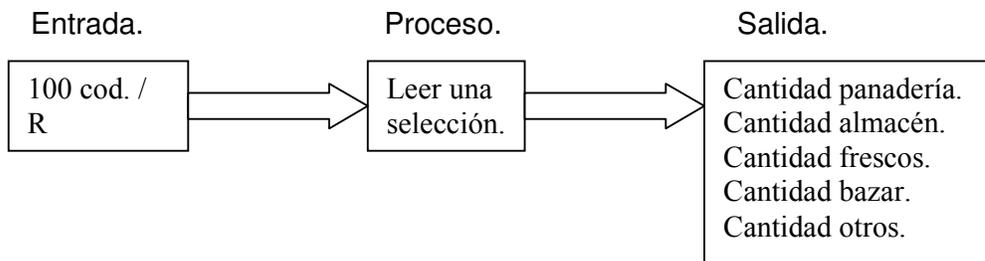
Ejemplo

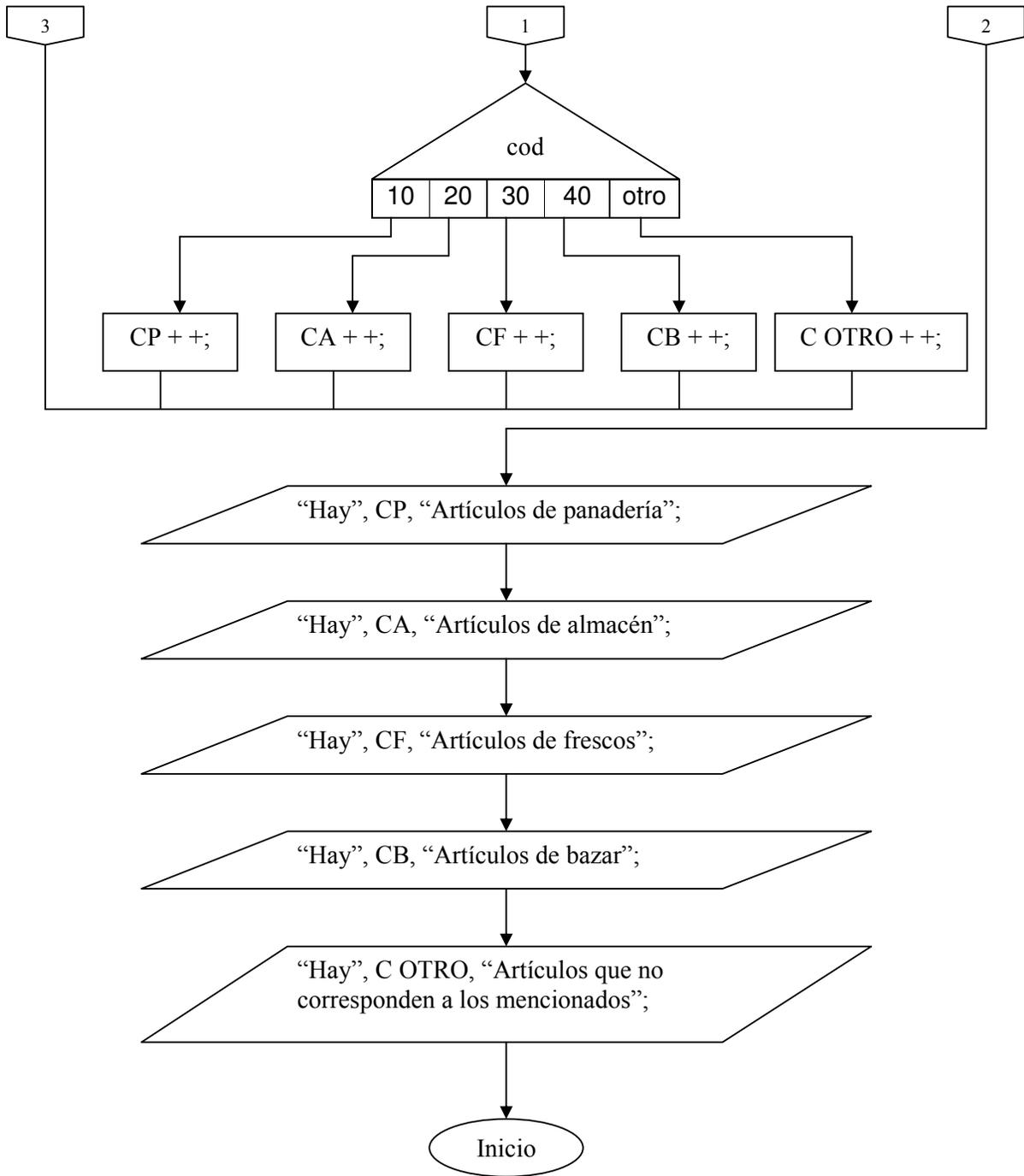
```
switch (expresion)
{
    case expresion_cte_1:
        sentencia_1;
        break;
    case expresion_cte_2: case expresion_cte_3:
        sentencia_2;
        break;
    default:
        sentencia_3;
}
```

Diseñar un programa en diagrama de flujo y pseudocódigo que determine la cantidad de artículos que pertenece a cada uno de los siguientes rubros en un mini mercado. Y cuantos de estos productos no pertenecen a ningún rubro antes mencionado.

La cantidad de artículos a ingresar en este caso es de 100 artículos.

Código / Rubro
10 → Panadería;
20 → almacén;
30 → frescos;
40 → bazar.





Pseudocódigo.

Inicio

```
{Imp "Este programa muestra la cantidad de articulos por rubros hasta llegar a mil.";
Para (I=1 , I<=1000 , I++ ;)
    {Imp "Ingrese código del articulo", I;
    Leer cod;
    Según (cod);
        { caso 10: CP ++;
        salto;
        caso 20: CA ++;
        salto;
        caso 30: CF ++;
        salto;
        caso 40: CB ++;
        salto;
        caso OTRO: C OTRO ++;
        }
    }
Imp "Hay", CP, "Artículos de panadería";
Imp "Hay", CA, "Artículos de almacén";
Imp "Hay", CF, "Artículos de frescos";
Imp "Hay", CB, "Artículos de bazar";
Imp "Hay", C OTRO, "Artículos que no corresponden a los mencionados";
}
```

Fin

Código

```
#include <stdio.h>
#define tam 5
int main(int argc, char **argv){
    int i=0,cod=0,CP=0,CA=0,CF=0,CB=0,C_OTRO=0;
    printf("Este programa muestra la cantidad de artículos por rubro hasta llegar a
    %d\n\n",tam);
    for(i=1;i<=tam;i++){
        printf("Descripción\tCódigo\n");
        printf("Panadería\t10\n");
        printf("Almacen\t20\n");
        printf("Frescos\t30\n");
        printf("Bazar\t40\n\n");
        printf("Ingrese el código del artículo %d: ",i);
        scanf("%d", &cod);
        switch(cod){
            case 10: CP ++;
                break;
            case 20: CA++;
                break;
            case 30: CF++;
                break;
            case 40: CB++;
                break;
        }
    }
}
```

```

        default: C_OTRO++;
    }
    printf("Hay %d artículos de panadería\n",CP);
    printf("Hay %d artículos de almacén\n",CA);
    printf("Hay %d artículos de frescos\n",CF);
    printf("Hay %d artículos de bazar\n",CB);
    printf("Hay %d artículos que no corresponden a los
mencionados\n\n",C_OTRO);
}
return 0;
}

```

Terminal

Este programa muestra la cantidad de artículos por rubro hasta llegar a 5

Descripción	Código
Panadería	10
Almacén	20
Frescos	30
Bazar	40

Ingrese el código del artículo 1: 10

Hay 1 artículos de panadería

Hay 0 artículos de almacén

Hay 0 artículos de frescos

Hay 0 artículos de bazar

Hay 0 artículos que no corresponden a los mencionados

Descripción	Código
Panadería	10
Almacén	20
Frescos	30
Bazar	40

Ingrese el código del artículo 2: 20

Hay 1 artículos de panadería

Hay 1 artículos de almacén

Hay 0 artículos de frescos

Hay 0 artículos de bazar

Hay 0 artículos que no corresponden a los mencionados

Descripción	Código
Panadería	10
Almacén	20
Frescos	30
Bazar	40

Ingrese el código del artículo 3: 30

Hay 1 artículos de panadería

Hay 1 artículos de almacén

Hay 1 artículos de frescos

Hay 0 artículos de bazar

Hay 0 artículos que no corresponden a los mencionados

Descripción	Código
Panadería	10
Almacén	20
Frescos	30
Bazar	40

Ingrese el código del artículo 4: 60

Hay 1 artículos de panadería

Hay 1 artículos de almacén

Hay 1 artículos de frescos

Hay 0 artículos de bazar

Hay 1 artículos que no corresponden a los mencionados

Descripción	Código
Panadería	10
Almacén	20
Frescos	30
Bazar	40

Ingrese el código del artículo 5: 70

Hay 1 artículos de panadería

Hay 1 artículos de almacén

Hay 1 artículos de frescos

Hay 0 artículos de bazar

Hay 2 artículos que no corresponden a los mencionados

(program exited with code: 0)

Press return to continue

Trabajo Practico numero 6

Actividad 1: Codificar en C todos los diagramas de flujo realizados en clase práctica y teórica que usen estructura selectiva múltiple.

Actividad 2: Correr todos los ejemplos realizados en clase teórica y práctica con estructura selectiva múltiple en el Laboratorio de informática modificar las salidas e implementar mejoras si es posible.

Actividad 3: Diseñar un programa en diagrama de flujo y codifíquelo en C, que determine lo siguiente: En una localidad se quiere saber cuántos alumnos están cursando cada uno de los seis años del secundario, para ello se cuenta con la siguiente información: un listado donde figura curso, sección, colegio y cantidad de alumnos. El fin del proceso será detectado, cuando curso sea igual a cero.

Actividad 4: Diseñar un programa en diagrama de flujo y codifíquelo en C, que simule un calculador de bolsillo. Que pueda sumar restar multiplicar y dividir dos números ingresados por teclado y que muestre un menú con estas opciones mas la opción salir del programa

Actividad 5: Ídem al anterior pero el programa deberá seleccionar la operación según el operador ingresado (+, *, /, -)

Actividad 6: Modificar el ejemplo del mini mercado para que se termine el proceso cuando el operador decida .Además los gerentes de almacén y bazar desean armar islas de liquidación de mercadería todo por cinco pesos en almacén y todo por diez pesos en bazar y para ello quieren saber cuantos artículos cuestan ese valor mas menos un diez por ciento y un listado de los mismos .Para ello cuentan con un listado de las con todos los artículos del súper con su código ,código de rubro ,importe, descripción ,cantidad en stock y cantidad en góndola .