
# Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño


ESTRUCTURAS

**LENGUAJE C**

PRÁCTICA 8

**Ingeniero en Software y Tecnologías Emergentes**

Brayan Ivan Perez Ventura

372781

22 de Noviembre del 2023 en Ensenada, Baja California

## Práctica 7. Estructuras

Desarrollen el código en lenguaje C y elaboren el diagrama de flujo correspondiente para los ejercicios. Será suficiente con un archivo .cpp que contenga todos los ejercicios organizados en un menú.

## Repositorio:

**PVBI_Lenguaje_C_932/Practicas/Practica8_Estructuras_PerezVentura_BrayanIvan at main · keiwv/PVBI_Lenguaje_C_932 (github.com)**

## DECLARACIÓN DE LIBRERÍAS, CONSTANTES, ESTRUCTURAS Y FUNCIONES PROTOTIPO

Para comenzar, se declararon las librerías a utilizar, que en este caso serían 2, una personal llamada “Frijoles.h” y otra llamada “time.h” (para sembrar una semilla generadora de números aleatorios). Enseguida, se declaró una constante llamada “MAX_REGISTERS” donde, este valor es el máximo registros permitidos por el arreglo.

Enseguida, se crea una estructura el cual almacenará el status del producto, el nombre, la cantidad, el precio y un código de barras para identificar el producto.

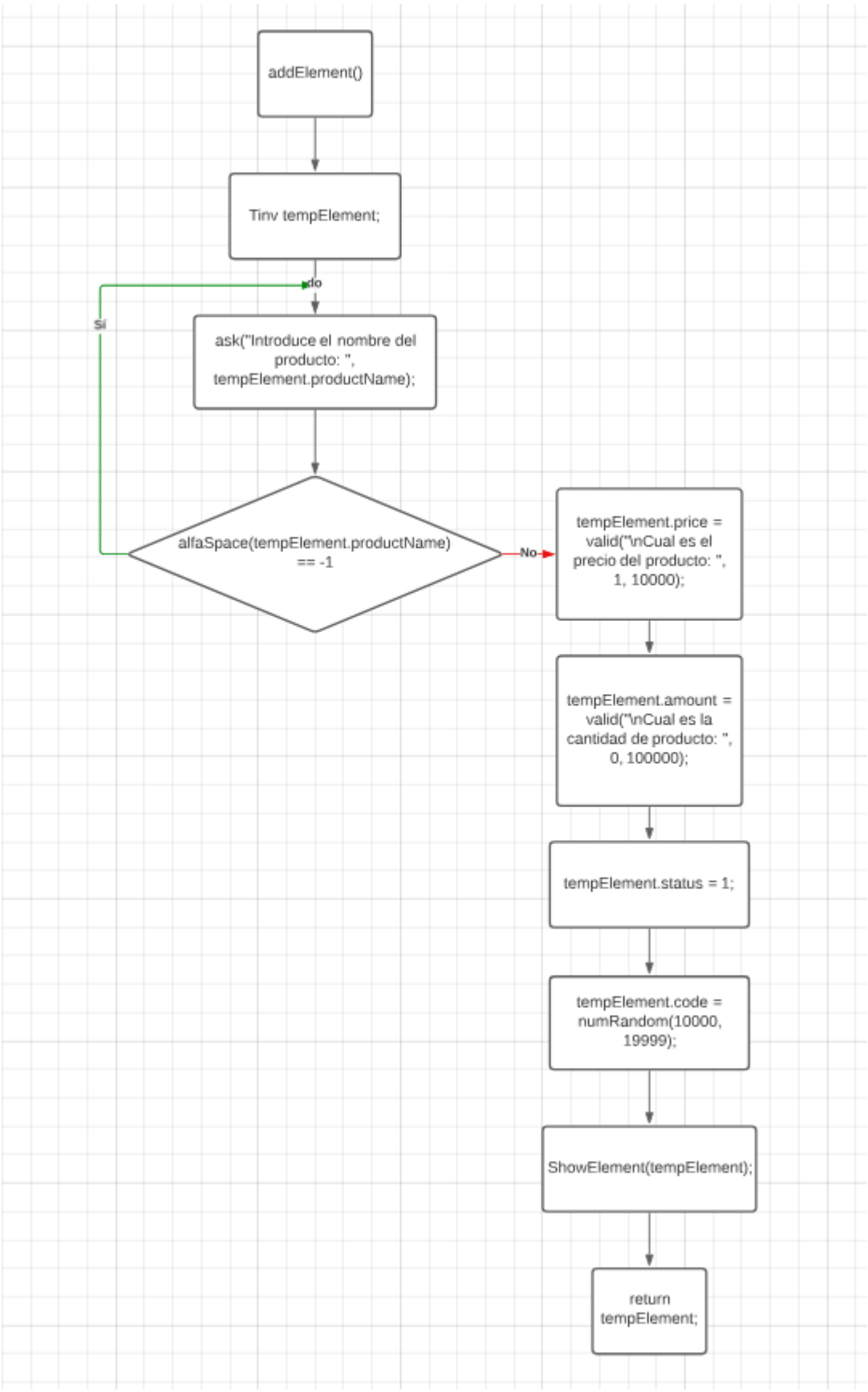
Finalmente, se declaran prototipos de funciones a utilizar.

```
12  #include "Frijoles.h"
13  #include <time.h>
14
15  #define MAX_REGISTERS 500
16  // *** STRUCTURES **
17  typedef struct _Inv
18  {
19      int status;
20      char productName[30];
21      int amount;
22      float price;
23      int code;
24  } Tinv;
25
26  //*** PROTOTYPE FUNCTIONS *******
27  void menu();
28  int msge_menu();
29
30  Tinv addElement();
31  void dltElement(Tinv products[], int position);
32  void ShowInventory(Tinv products[], int position);
33  void calculateValueInventory(Tinv products[], int position);
34
35  void ShowElement(Tinv product);
36  int existElem(Tinv products[], int longi, int num);
37
```

## AGREGAR ELEMENTOS

En la función, agregar elementos, se le preguntará al usuario los datos de los productos, como el usuario está registrando dicho producto, se considerará como el “status activo (1)” y se le agregará un código para identificarlo aleatoriamente.

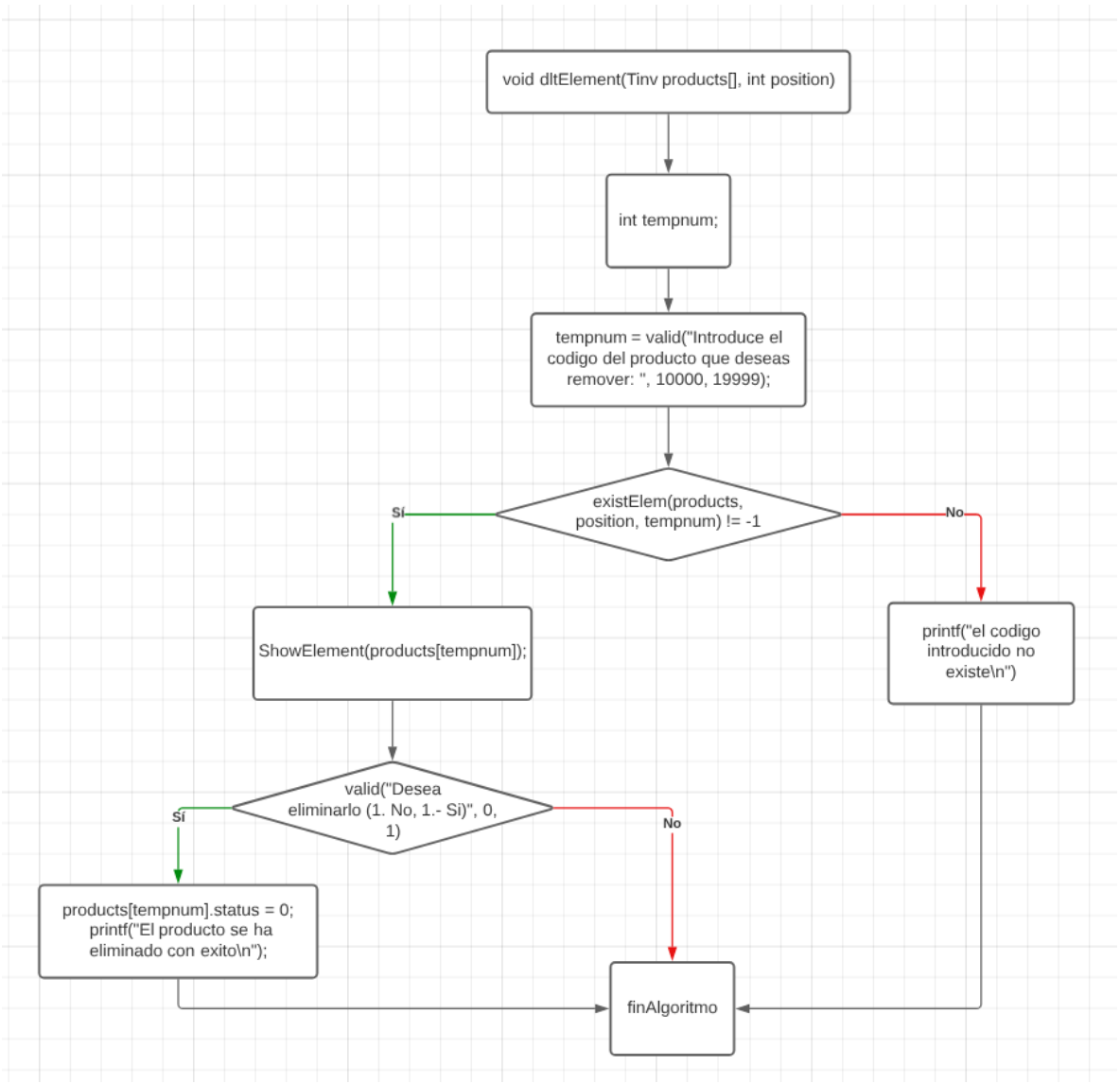
```
Tinv addElement()
{
    Tinv tempElement;
    do
    {
        ask("Introduce el nombre del producto: ", tempElement.productName);
    } while (alfaSpace(tempElement.productName) == -1);

    tempElement.price = valid("\nCual es el precio del producto: ", 1, 10000);

    tempElement.amount = valid("\nCual es la cantidad de producto: ", 0, 100000);

    tempElement.status = 1;

    tempElement.code = numRandom(10000, 19999);

    ShowElement(tempElement);
    return tempElement;
}
```


addElement()
Tinv tempElement;
do
ask("Introduce el nombre del producto: ", tempElement.productName);
alfaSpace(tempElement.productName) == -1
Si
No
tempElement.price = valid("\nCual es el precio del producto: ", 1, 10000);
tempElement.amount = valid("\nCual es la cantidad de producto: ", 0, 100000);
tempElement.status = 1;
tempElement.code = numRandom(10000, 19999);
ShowElement(tempElement);
return tempElement;

## ELIMINAR ELEMENTOS

Para el de eliminar elementos, se le preguntará al usuario por el código identificador del producto que busca, en caso de que no exista, se le desplegará un mensaje “El código introducido no existe”, en de que sí exista, se le desplegará el producto con el precio, cantidad, nombre, código, para así preguntarle si lo quiere eliminar o no.

```
135 void dltElement(Tinv products[], int position)
136 {
137     int tempnum;
138 
139     tempnum = valid("Introduce el codigo del producto que deseas remover: ", 10000, 19999);
140 
141     if (existElem(products, position, tempnum) != -1)
142     {
143         ShowElement(products[tempnum]);
144         if (valid("Desea eliminarlo (1. No, 1.- Si)", 0, 1))
145         {
146             products[tempnum].status = 0;
147             printf("El producto se ha eliminado con exito\n");
148         }
149     }
150     else
151     {
152         printf("El codigo introducido no existe\n");
153     }
154 }
```

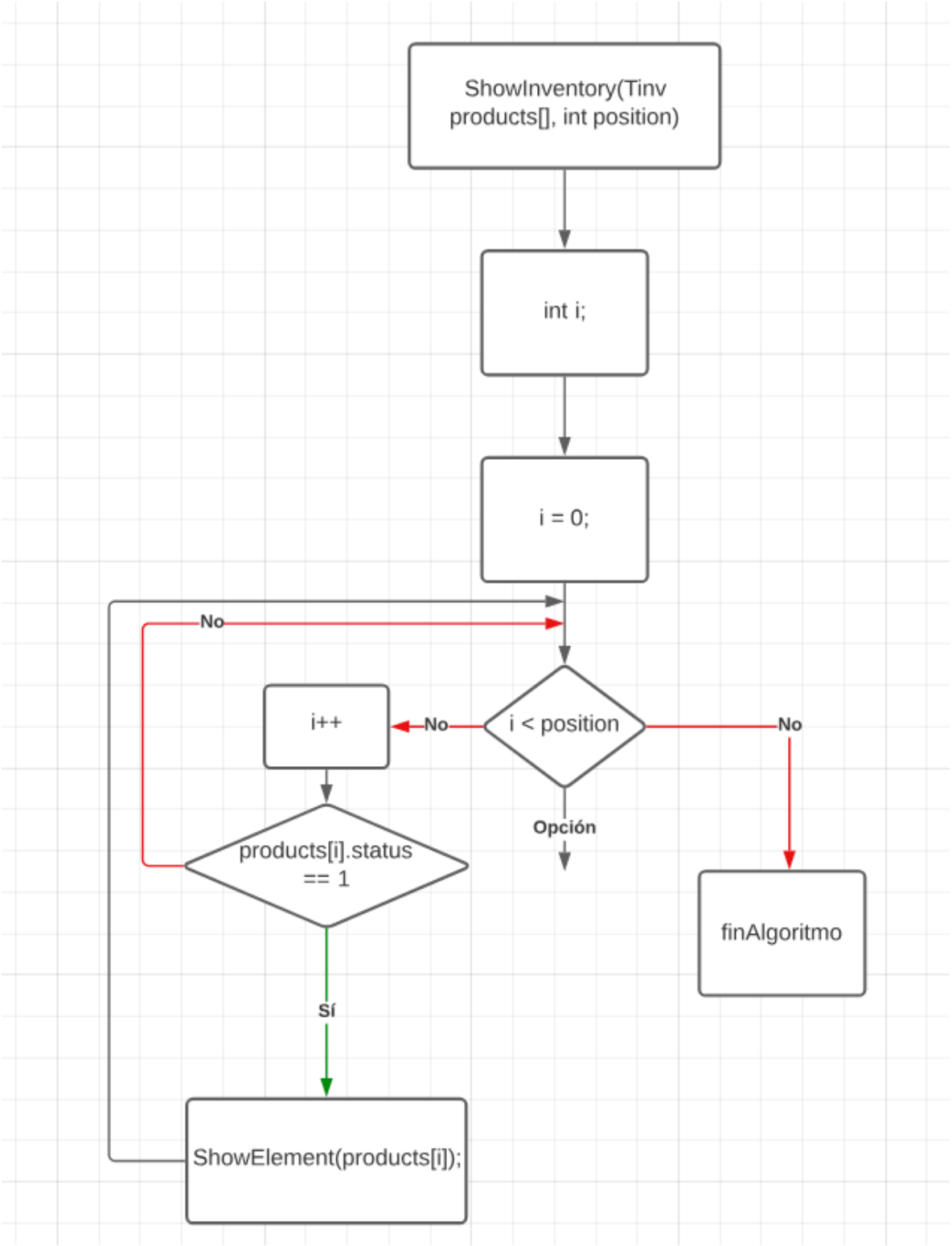

void dltElement(Tinv products[], int position)
int tempnum;
tempnum = valid("Introduce el codigo del producto que deseas remover: ", 10000, 19999);
existElem(products, position, tempnum) != -1
Sí
No
ShowElement(products[tempnum]);
printf("el codigo introducido no existe\n")
valid("Desea eliminarlo (1. No, 1.- Si)", 0, 1)
Sí
No
products[tempnum].status = 0;
printf("El producto se ha eliminado con exito\n");
finAlgoritmo

## MOSTRAR ELEMENTOS

Para esta función, únicamente se crea un for que llegue hasta la posición en el que está actualmente la cantidad de productos.

Si la condición de que el status sea activo, este se desplegará. Sucederá para todos los productos registrados.

```
164  void ShowInventory(Tinv products[], int position)
165  {
166      int i;
167      for (i = 0; i < position; i++)
168      {
169          if (products[i].status == 1)
170          {
171              ShowElement(products[i]);
172          }
173      }
174  }
```


ShowInventory(Tinv products[], int position)
int i;
i = 0;
No
i < position
No
No
i++
products[i].status == 1
Opción
finAlgoritmo
Sí
ShowElement(products[i]);

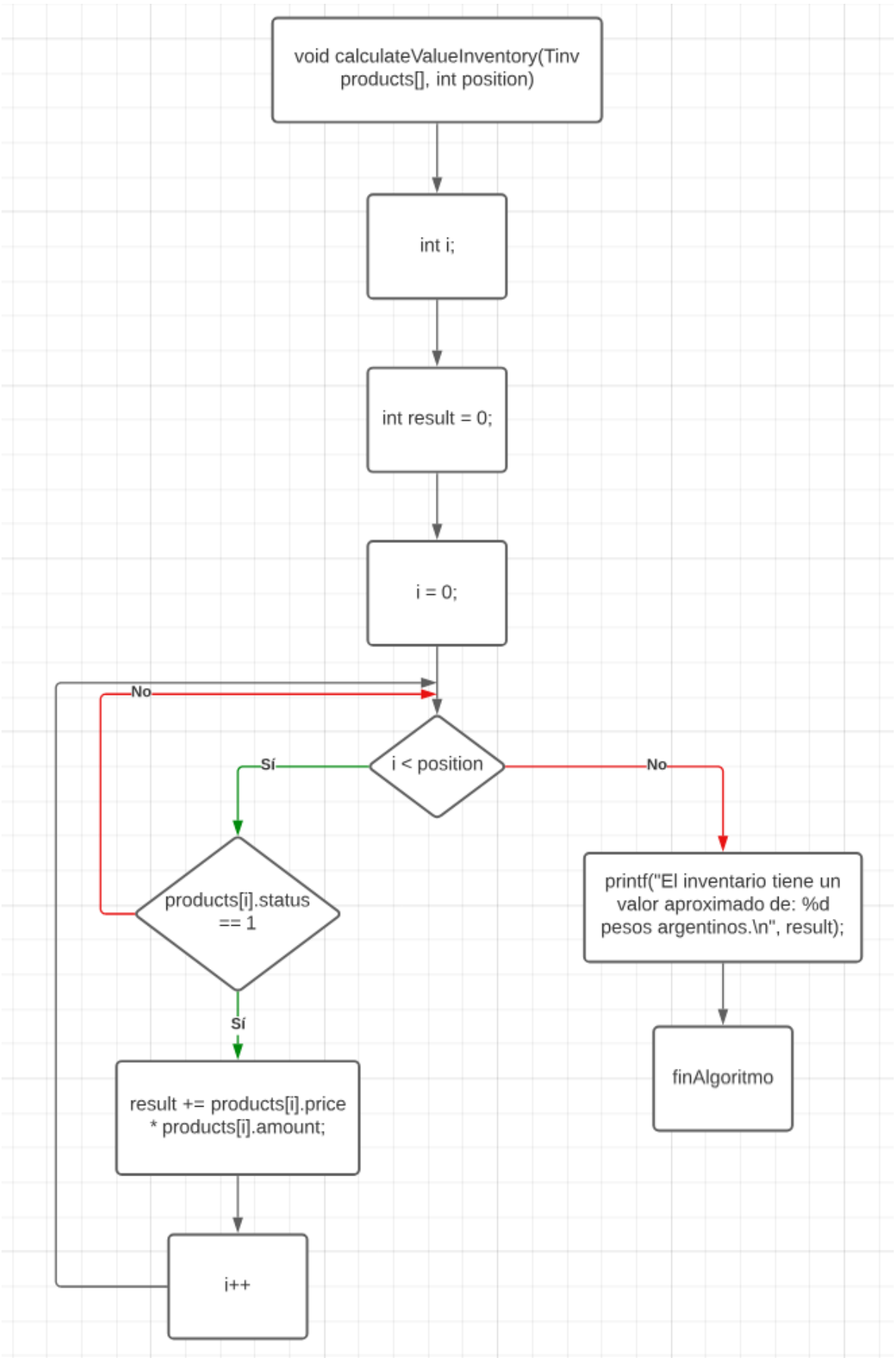
## CALCULAR EL VALOR DEL INVENTARIO

En esta función, se implementa un for que llega hasta la posición en la que está actualmente el arreglo.

Si el status es activo, este sumará la multiplicación de la cantidad de productos por el precio del producto.

Finalmente se imprimirá el resultado obtenido.

```
184  void calculateValueInventory(Tinv products[], int position)
185  {
186      int i;
187      int result = 0;
188      for (i = 0; i < position; i++)
189      {
190          if (products[i].status == 1)
191          {
192              result += products[i].price * products[i].amount;
193          }
194      }
195      printf("El inventario tiene un valor aproximado de: %d pesos argentinos.\n", result);
196  }
```


void calculateValueInventory(Tinv products[], int position)
int i;
int result = 0;
i = 0;
No
i < position
Sí
No
products[i].status == 1
Sí
result += products[i].price * products[i].amount;
i++
printf("El inventario tiene un valor aproximado de: %d pesos argentinos.\n", result);
finAlgoritmo