



Facultad de Ciencia, Tecnología y Ambiente

INGENIERÍA EN SISTEMAS DE INFORMACIÓN
Metodología de la Programación

Docente:

Ing. José Alejandro Duran García

PRÁCTICA DE FAMILIARIZACIÓN

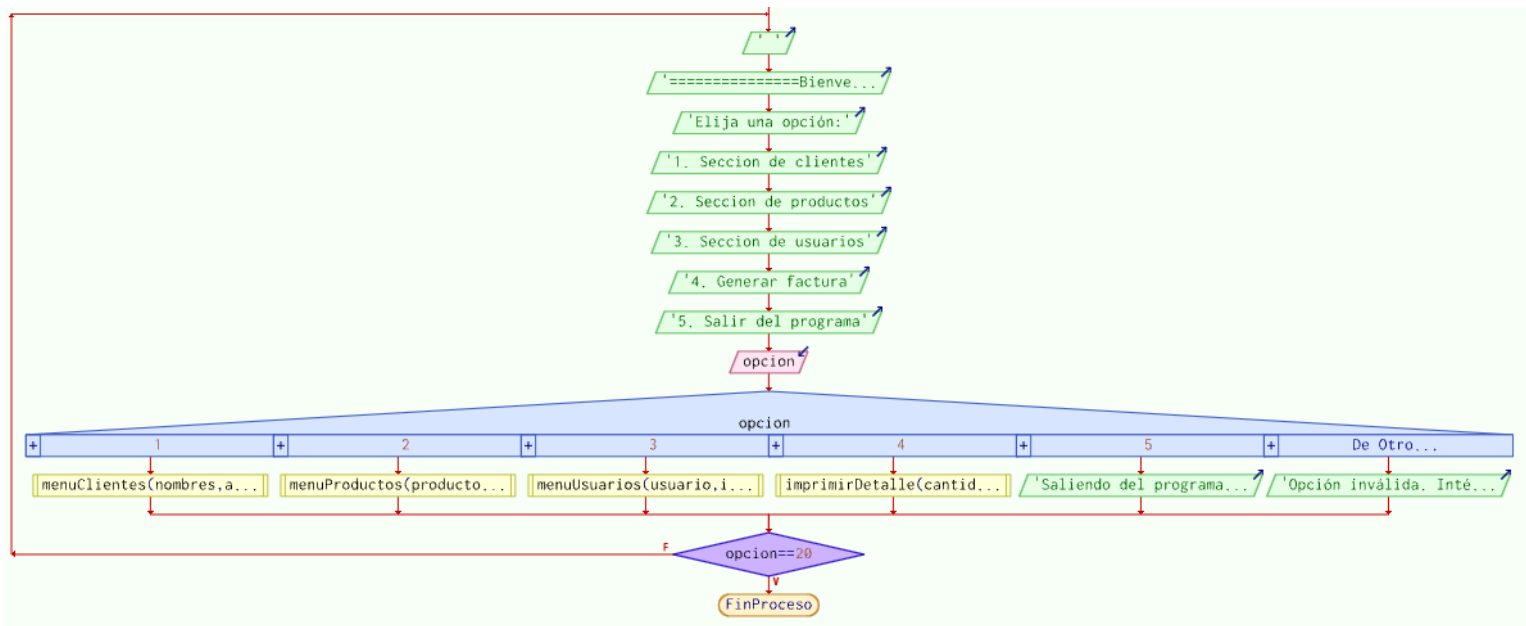
Autores:

1. Diego Francisco Mora Castillo
2. Carlos Francisco Talavera Guerrero
3. Enrique José Taleno Nuñez
4. Leandro Núñez Duarte

Managua, Nicaragua - 2023

Contenido

1. FASE DE ANÁLISIS	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. DIAGRAMA DE ESTRUCTURA	1
1.3. RESULTADOS DE LA FASE DE ANÁLISIS	2
2. FASE DE DISEÑO.....	6
1.1. ALGORITMOS	6
1.1.1. Pseudocódigo	6
1.1.2. Diagrama de Flujo de Datos	19
3. FASE DE CODIFICACIÓN Y DEPURACIÓN	38
3.1. Estructuras de datos del proyecto	38
3.2. Código fuente y salidas	40



1.3. RESULTADOS DE LA FASE DE ANÁLISIS

DATOS DE ENTRADA y SALIDA			
Identificador	Tipo de entrada	Tipo de dato	Valor inicial
ID	Desconocido	Carácter	0
nombre	Desconocido	Carácter	0
apellido	Desconocido	Carácter	0
IDP	Desconocido	Carácter	0
nombreP	Conocido	Carácter	0
descripcion	Conocido	Carácter	0
precio	Conocido	Decimal	0
IDU	Desconocido	Carácter	0
nombreU	Desconocido	Carácter	0
email	Desconocido	Carácter	0
password	Desconocido	Carácter	0
numFactura	Desconocido	Entero	0
fecha	Conocido	Carácter	0
nombreCliente	Desconocido	Carácter	0
observacion	Conocido	Carácter	0
idDetalle	Desconocido	Entero	0
numFac	Desconocido	Entero	0
nomProducto	Conocido	Carácter	0
precioP	Conocido	Decimal	0

cantidad	Conocido	Entero	0
----------	----------	--------	---

MÉTODOS		
Nombre	Descripción	Fórmulas
agregarCliente	Función para agregar la identificación del cliente, nombre, apellido	Los datos de los clientes son almacenados en la última posición en una variable de una estructura exclusiva para los clientes, luego estos datos se guardan en otra variable del mismo tipo de estructura, después se suma la última posición en uno en uno cada vez que se ejecute
mostrarCliente	Función para mostrar los datos de un cliente a la vez	Imprime en pantalla los datos almacenados de un cliente específico
mostrarClientes	Función que nos permitirá visualizar los clientes	Evalúa si se ha agregado algún cliente, si no se ha agregado ningún cliente se imprime un mensaje que dice que no hay registro; En caso que haya un cliente registrado se ejecuta un ciclo que muestra todos los clientes registrados hasta el momento
"case 2" de la sección de clientes en el menú MenuCompleteClient	Serie de procesos para editar los datos del cliente seleccionado	Se piden los datos a editar del cliente después busca a ese cliente específico luego se muestra al cliente buscado para editar los datos de ese cliente y reemplazarlos
borrarClientes	Función para cambiar la posición del cliente para luego ser borrado	Evalúa si se ha agregado algún cliente, si no se ha agregado ningún cliente se imprime un mensaje que dice que no hay registro; en caso de que haya un cliente registrado se ejecuta un ciclo que mueve al cliente hasta el penúltimo registro y después decrementa el valor del último registro para luego usar la función ClienteInicial

agregarProductos	Función que nos permitirá añadir un producto al inventario	Los datos de los productos son almacenados en la última posición en una variable de una estructura exclusiva para los productos, luego estos datos se guardan en otra variable del mismo tipo de estructura, después se suma la última posición en uno en uno cada vez que se ejecute
mostrarProducto	Función para mostrar los datos de un producto a la vez	Imprime en pantalla los datos almacenados de un producto específico
mostrarProductos	Función que nos permitirá visualizar los productos	Evalúa si se ha agregado algún producto, si no se ha agregado ningún producto se imprime un mensaje que dice que no hay registro; En caso que haya un producto registrado se ejecuta un ciclo que muestra todos los productos registrados hasta el momento
"case 2" de la sección de productos en el menú MenuCompleteProduct	Serie de procesos para editar los datos del producto seleccionado	Se piden los datos a editar del producto después busca a ese producto específico luego se muestra al producto buscado para editar los datos de ese producto y reemplazarlos
borrarProducto	Función que nos permitirá borrar un producto seleccionado	Evalúa si se ha agregado algún producto, si no se ha agregado ningún producto se imprime un mensaje que dice que no hay registro; en caso de que haya un producto registrado se ejecuta un ciclo que mueve al producto hasta el penúltimo registro y después decrementa el valor del último registro para luego usar la función ProductoInicial
agregarUsuario	Función que nos permitirá agregar un usuario al sistema	Los datos de los usuarios son almacenados en la última posición en una variable de una estructura exclusiva para los usuarios, luego estos datos se guardan en otra variable del mismo tipo de estructura, después se suma la última posición en uno en uno cada vez que se ejecute

mostrarUsuario	Función para mostrar los datos de un usuario a la vez	Imprime en pantalla los datos almacenados de un usuario específico
mostrarUsuarios	Función que nos permitirá visualizar los usuarios	Evalúa si se ha agregado algún usuario, si no se ha agregado ningún usuario se imprime un mensaje que dice que no hay registro; En caso que haya un producto registrado se ejecuta un ciclo que muestra todos los productos registrados hasta el momento
"case 2" de la sección de productos en el menú MenuCompleteUsuario	Serie de procesos para editar los datos del usuario seleccionado	Se piden los datos a editar del usuario después busca a ese usuario específico luego se muestra al usuario buscado para editar los datos de ese usuario y reemplazarlos
borrarUsuario	Función para borrar al Usuario seleccionado	Evalúa si se ha agregado algún usuario, si no se ha agregado ningún producto se imprime un mensaje que dice que no hay registro; en caso de que haya un usuario registrado se ejecuta un ciclo que mueve al usuario hasta el penúltimo registro y después decrementa el valor del último registro para luego usar la función InicialUsuario

SALIDA 1
<p>Agregando clientes... ID del cliente: Nombre del cliente: Apellido del cliente:</p> <p>Agregando productos... ID del producto: Nombre del producto: Descripción del producto: Precio del producto:</p> <p>Agregando usuarios... ID de usuario: Nombre de usuario: Email de usuario:</p>

Contraseña del usuario:

Generando factura...

```
Ingresa la fecha: 6/6/2023
Ingresa el cliente: Diego
Ingresa una observacion: Completado
Cuantos productos va a facturar?: 2
Ingresa el codigo del producto: 01
Producto: Camisa - Precio: 30.00
Ingresa la cantidad de productos: 2
Ingresa el codigo del producto: 05
Producto: Camisa Talla X - Precio: 130.00
Ingresa la cantidad de productos: 1
=====MCSerigraph=====
Direccion: Masaya
Telefono: 25613023
Factura #: 4
Fecha: 6/6/2023
=====
Cliente: Diego
PRODUCTO          | CANTIDAD          | PRECIO            | TOTAL
Camisa             | 2                 | 30.00             | 60.00
Camisa Talla X     | 1                 | 130.00            | 130.00
Monto a pagar: 190.00
=====
Gracias por su compra
=====
Presione una tecla para continuar . . .
```

2. FASE DE DISEÑO

1.1. ALGORITMOS

1.1.1. Pseudocódigo

Sub Procesos agregarClientes y mostrarClientes

SubProceso agregarClientes(nombres, apellidos, ids, cantidad_clientes)

 Escribir "Ingrese el nombre del cliente:";

 Leer nombres[cantidad_clientes];

 Escribir "Ingrese el apellido del cliente:";

 Leer apellidos[cantidad_clientes];

 Escribir "Ingrese el ID del cliente:";

 Leer ids[cantidad_clientes];

 cantidad_clientes <- cantidad_clientes + 1;

 Escribir "Cliente agregado correctamente.";

FinSubProceso

SubProceso mostrarClientes(nombres, apellidos, ids, cantidad_clientes, i)

 Si cantidad_clientes == 0 Entonces

 Escribir "No hay clientes registrados.";

 Sino

 Escribir "Clientes registrados:";

 Para i <- 0 Hasta cantidad_clientes-1 Con Paso 1 Hacer

 Escribir "Nombre: ", nombres[i];

 Escribir "Apellido: ", apellidos[i];

 Escribir "ID: ", ids[i];

 Escribir "";

 FinPara

 FinSi

FinSubProceso

Sub Procesos editarClientes y eliminarCliente

SubProceso editarCliente(nombres, apellidos, ids, cantidad_clientes, id_cliente)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del cliente a editar:";

Leer id_cliente[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_clientes-1 Con Paso 1 Hacer

Si ids[i] == id_cliente[contador1] Entonces

Escribir "Ingrese el nuevo nombre del cliente:";

Leer nombres[i];

Escribir "Ingrese el nuevo apellido del cliente:";

Leer apellidos[i];

Escribir "Ingrese el nuevo ID del cliente:";

Leer ids[i];

Encontrado <- Verdadero;

Escribir "Cliente editado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró un cliente con el ID especificado.";

FinSi

FinSubProceso

SubProceso eliminarCliente(nombres, apellidos, ids, cantidad_clientes,
id_cliente)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del cliente a eliminar:";

Leer id_cliente[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_clientes-1 Con Paso 1 Hacer

Si ids[i] == id_cliente[contador1] Entonces

nombres[i] <- "";

apellidos[i] <- "";

ids[i] <- "";

Encontrado <- Verdadero;

cantidad_clientes <- cantidad_clientes - 1;

Escribir "Cliente eliminado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró un cliente con el ID especificado.";

FinSi

FinSubProceso

Sub Procesos agregarProducto y mostrarProductos

SubProceso agregarProducto(producto, cantidad, idProducto, descripcion, precio, cantidad_productos)

```
    Escribir "Ingrese el nombre del producto:";
    Leer producto[cantidad_productos];
    Escribir "Ingrese la descripcion del producto:";
    Leer descripcion[cantidad_productos];
    Escribir "Ingrese el ID del producto:";
    Leer idProducto[cantidad_productos];
    Escribir "Ingrese la cantidad del producto:";
    Leer cantidad[cantidad_productos];
    Escribir "Dime el precio del producto:";
    Leer precio[cantidad_productos];
    cantidad_productos <- cantidad_productos + 1;
    Escribir "Producto agregado correctamente.";
```

FinSubProceso

SubProceso mostrarProductos(producto, cantidad, idProducto, descripcion, precio, cantidad_productos, i)

```
    Si cantidad_productos == 0 Entonces
        Escribir "No hay productos registrados.";
    Sino
        Escribir "Productos registrados:";
        Para i <- 0 Hasta cantidad_productos-1 Con Paso 1 Hacer
            Escribir "Nombre del producto: ", producto[i];
            Escribir "Descripcion del producto: ", descripcion[i];
            Escribir "ID del producto: ", idProducto[i];
            Escribir "Cantidad: ", cantidad[i];
            Escribir "Precio:", precio[i];
            Escribir "";
```

FinPara

FinSi

FinSubProceso

Sub Procesos editarProducto y eliminarProducto

SubProceso editarProducto(producto, cantidad, idProducto, descripcion, precio, cantidad_productos, id_producto)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del producto a editar:";

Leer id_producto[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_productos-1 Con Paso 1 Hacer

Si idProducto[i] == id_producto[contador1] Entonces

Escribir "Ingresa el nuevo nombre del producto:";

Leer producto[i];

Escribir "Ingrese la nueva descripcion del producto:";

Leer descripcion[i];

Escribir "Ingrese el nuevo ID del producto:";

Leer idProducto[i];

Escribir "Ingrese la nueva cantidad de producto:";

Leer cantidad[i];

Escribir "Ingrese el nuevo precio del producto:";

Leer precio[i];

Encontrado <- Verdadero;

Escribir "Producto eliminado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró el producto con el ID especificado.";

FinSi

FinSubProceso

SubProceso eliminarProducto(producto, cantidad, idProducto, descripcion, precio, cantidad_productos, id_producto)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del producto a eliminar:";

Leer id_producto[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_productos-1 Con Paso 1 Hacer

Si idProducto[i] == id_producto[contador1] Entonces

producto[i] <- "";

descripcion[i] <- "";

idProducto[i] <- "";

cantidad[i] <- "";

precio[i] <- "";

Encontrado <- Verdadero;

cantidad_productos <- cantidad_productos - 1;

Escribir "Producto eliminado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró el producto con el ID especificado.";

FinSi

FinSubProceso

Sub Procesos agregarUsuario y mostrarUsuarios

SubProceso agregarUsuario(usuario, idUsuario, password, cantidad_usuarios)

```
    Escribir "Nombre del usuario: ";
    Leer usuario[cantidad_usuarios];
    Escribir "Ingrese su ID: ";
    Leer idUsuario[cantidad_usuarios];
    Escribir "Ingrese su contraseña";
    Leer password[cantidad_usuarios];
    Escribir "Ingrese la cantidad de usuarios";
    cantidad_usuarios <- cantidad_usuarios + 1;
    Escribir "Usuario agregado correctamente.";
```

FinSubProceso

SubProceso mostrarUsuarios(usuario, idUsuario, password, cantidad_usuarios, i)

```
    Si cantidad_usuarios == 0 Entonces
        Escribir "No hay usuario registrados.";
    Sino
        Escribir "Usuario registrados:";
        Para i <- 0 Hasta cantidad_usuarios-1 Con Paso 1 Hacer
            Escribir "Nombre de usuario: ", usuario[i];
            Escribir "ID del usuario: ", idUsuario[i];
            Escribir "Contraseña: ", password[i];
            Escribir "";
```

FinPara

FinSi

FinSubProceso

Sub Procesos editarUsuario y eliminarUsuario

SubProceso editarUsuario(usuario, idUsuario, password, cantidad_usuarios, id_usuario)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del usuario a editar:";

Leer id_usuario[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_usuarios-1 Con Paso 1 Hacer

Si idUsuario[i] == id_usuario[contador1] Entonces

Escribir "Ingresa el nuevo nombre de usuario:";

Leer usuario[i];

Escribir "Ingrese el nuevo ID de usuario:";

Leer idUsuario[i];

Escribir "Ingrese la password de usuario:";

Leer password[i];

Encontrado <- Verdadero;

Escribir "Usuario editado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró el usuario con el ID especificado.";

FinSi

FinSubProceso

SubProceso eliminarUsuario(usuario, idUsuario, password, cantidad_usuarios, id_usuario)

Definir i Como Entero;

Definir Encontrado Como Logico;

Definir contador1 Como Entero;

contador1<-1;

Escribir "Ingrese el ID del usuario a eliminar:";

Leer id_usuario[contador1];

Encontrado <- Falso;

Para i <- 0 Hasta cantidad_usuarios-1 Con Paso 1 Hacer

Si idUsuario[i] == id_usuario[contador1] Entonces

usuario[i] <- "";

idUsuario[i] <- "";

password[i] <- "";

Encontrado <- Verdadero;

cantidad_usuarios <- cantidad_usuarios - 1;

Escribir "Usuario eliminado correctamente.";

FinSi

FinPara

Si Encontrado == Falso Entonces

Escribir "No se encontró el producto con el ID especificado.";

FinSi

FinSubProceso

Sub Procesos encabezadoFactura, pieDeFactura, imprimirDetalle

SubProceso encabezadoFactura(cantidad_facturas, cliente, fecha)

```
    Escribir "=====Bienvenido a
    MCSerigraph=====";
    Escribir "Direccion: Masaya";
    Escribir "Telefono: 25613023";
    Escribir "Factura #: ", cantidad_facturas;
    Escribir "Fecha: ", fecha[cantidad_facturas];
    Escribir
"=====";
    Escribir "Cliente: ", cliente[cantidad_facturas];
FinSubProceso
```

SubProceso pieDeFactura()

```
    Escribir
"=====";
    Escribir "Gracias por su compra";
    Escribir
"=====";
FinSubProceso
```

SubProceso imprimirDetalle(cantidad_facturas, cliente, fecha, cantidadTemporal, precioTemporal, montoTemporal, totalTemporal, producto, cantidad, idProducto, descripcion, precio, cantidad_productos, id_producto)

```
    Definir i Como Entero;
    Definir j Como Entero;
    Definir cantidadProductFactura Como Entero;
    cantidadProductFactura <- 0;
    Definir contador1 Como Entero;
    contador1<-1;

    Escribir "Ingresa la fecha: ";
    Leer fecha[cantidad_facturas];
    Escribir "Ingresa el cliente: ";
    Leer cliente[cantidad_facturas];

    encabezadoFactura(cantidad_facturas, cliente, fecha);

    Escribir "Cuantos productos desea facturar?";
    Leer cantidadProductFactura;
    Para j <- 0 Hasta cantidadProductFactura-1 Con Paso 1 Hacer
        Escribir "Ingrese el ID del producto:";
        Leer id_producto[contador1];
        Para i <- 0 Hasta cantidad_productos-1 Con Paso 1 Hacer
            Si idProducto[i] == id_producto[contador1] Entonces
                Escribir producto[i];
                Escribir precio[i];
            FinSi
        FinPara
    FinPara
    pieDeFactura();
FinSubProceso
```

SubProceso menuClientes nombres, apellidos, ids, cantidad_clientes, id_cliente, i

SubProceso menuClientes(nombres, apellidos, ids, cantidad_clientes, id_cliente, i)

```
    Definir opcion Como Entero;
    Repetir
        Escribir "=====Bienvenido a
MCSerigraph=====";
        Escribir "Seccion de clientes...";
        Escribir "Elija una opción:";
        Escribir "1. Agregar cliente";
        Escribir "2. Mostrar clientes";
        Escribir "3. Editar cliente";
        Escribir "4. Eliminar cliente";
        Escribir "5. Salir";
        Leer opcion;
        Segun opcion Hacer
            1:
                agregarClientes(nombres, apellidos, ids,
cantidad_clientes);
                cantidad_clientes <- cantidad_clientes + 1;
            2:
                mostrarClientes(nombres, apellidos, ids,
cantidad_clientes, i);
            3:
                editarCliente(nombres, apellidos, ids,
cantidad_clientes, id_cliente);
            4:
                eliminarCliente(nombres, apellidos, ids,
cantidad_clientes, id_cliente);
            5:
                Escribir "Saliendo...";
        De Otro Modo:
            Escribir "Opcion invalida. Intente de nuevo.";
        FinSegun
    Hasta Que opcion == 5;
FinSubProceso
```


**SubProceso menuProductos producto, cantidad, idProducto, descripcion,
precio, cantidad_productos, id_producto, i**

SubProceso menuProductos(producto, cantidad, idProducto, descripcion,
precio, cantidad_productos, id_producto, i)

Definir opcion Como Entero;

Repetir

Escribir "=====Bienvenido a

MCSerigraph=====";

Escribir "Seccion de productos...";

Escribir "Elija una opción:";

Escribir "1. Agregar producto";

Escribir "2. Mostrar productos";

Escribir "3. Editar producto";

Escribir "4. Eliminar producto";

Escribir "5. Salir";

Leer opcion;

Segun opcion Hacer

1:

agregarProducto(producto, cantidad, idProducto,
descripcion, precio, cantidad_productos);
cantidad_productos <- cantidad_productos + 1;

2:

mostrarProductos(producto, cantidad, idProducto,
descripcion, precio, cantidad_productos, i);

3:

editarProducto(producto, cantidad, idProducto,
descripcion, precio, cantidad_productos,
id_producto);

4:

eliminarProducto(producto, cantidad, idProducto,
descripcion, precio, cantidad_productos,
id_producto);

5:

Escribir "Saliendo...";

De Otro Modo:

Escribir "Opcion invalida. Intente de nuevo.";

FinSegun

**SubProceso menuUsuarios usuario, idUsuario, password,
cantidad_usuarios, id_usuario, i**

SubProceso menuUsuarios(usuario, idUsuario, password, cantidad_usuarios,
id_usuario, i)

Definir opcion Como Entero;

Repetir

 Escribir "=====Bienvenido a

MCSerigraph=====";

 Escribir "Seccion de usuarios...";

 Escribir "Elija una opción:";

 Escribir "1. Agregar usuario";

 Escribir "2. Mostrar usuarios";

 Escribir "3. Editar usuario";

 Escribir "4. Eliminar usuario";

 Escribir "5. Salir";

 Leer opcion;

 Segun opcion Hacer

 1:

 agregarUsuario(usuario, idUsuario, password,
cantidad_usuarios);

 cantidad_usuarios<-cantidad_usuarios+1;

 2:

 mostrarUsuarios(usuario, idUsuario, password,
cantidad_usuarios, i);

 3:

 editarUsuario(usuario, idUsuario, password,
cantidad_usuarios, id_usuario);

 4:

 eliminarUsuario(usuario, idUsuario, password,
cantidad_usuarios, id_usuario);

 5:

 Escribir "Saliendo...";

 De Otro Modo:

 Escribir "Opcion invalida. Intente de nuevo.";

 FinSegun

 Hasta Que opcion == 5;

FinSubProceso Hasta Que opcion == 5;

FinSubProceso

Proceso proyectoFinal

Proceso proyectoFinal

Definir nombres, apellidos, ids Como Caracter;

Dimension nombres[50], apellidos[50], ids[50];

Definir producto, cantidad, idProducto, descripcion, precio Como Caracter;

Dimension producto[50], cantidad[50], idProducto[50], descripcion[50], precio[50];

Definir id_cliente, id_producto, id_usuario Como Caracter;

Dimension id_cliente[50], id_producto[50], id_usuario[50];

Definir usuario, idUsuario, password Como Caracter;

Dimension usuario[50], idUsuario[50], password[50];

Definir cantidad_clientes, cantidad_productos, cantidad_usuarios, cantidad_facturas Como Entero;

Definir opcion Como Entero;

Definir fecha, cliente Como Caracter;

Dimension fecha[15], cliente[50];

Definir EncontradoDelCliente Como Logico;

Definir cantidadTemporal, precioTemporal, montoTemporal, totalTemporal Como Real;

cantidadTemporal <- 0;

precioTemporal <- 0;

montoTemporal <- 0;

totalTemporal <- 0;

cantidad_productos <- 0;

cantidad_clientes <- 0;

cantidad_usuarios <- 0;

cantidad_facturas <- 0;

opcion <- 0;

Definir i, j Como Entero;

i <- 0;

j <- 0;

Proceso proyectoFinal CONTINUACIÓN

Repetir

 Escribir " ";

 Escribir "=====Bienvenido a

MCSerigraph=====";

 Escribir "Elija una opción:";

 Escribir "1. Seccion de clientes";

 Escribir "2. Seccion de productos";

 Escribir "3. Seccion de usuarios";

 Escribir "4. Generar factura";

 Escribir "5. Salir del programa";

 Leer opcion;

 Segun opcion Hacer

 1:

 menuClientes(nombres, apellidos, ids,
 cantidad_clientes, id_cliente, i);

 2:

 menuProductos(producto, cantidad, idProducto,
 descripcion, precio, cantidad_productos,
 id_producto, i);

 3:

 menuUsuarios(usuario, idUsuario, password,
 cantidad_usuarios, id_usuario, i);

 4:

 imprimirDetalle(cantidad_facturas, cliente, fecha,
 cantidadTemporal, precioTemporal,
 montoTemporal, totalTemporal, producto,
 cantidad, idProducto, descripcion, precio,
 cantidad_productos, id_producto);

 5:

 Escribir "Saliendo del programa...";

 De Otro Modo:

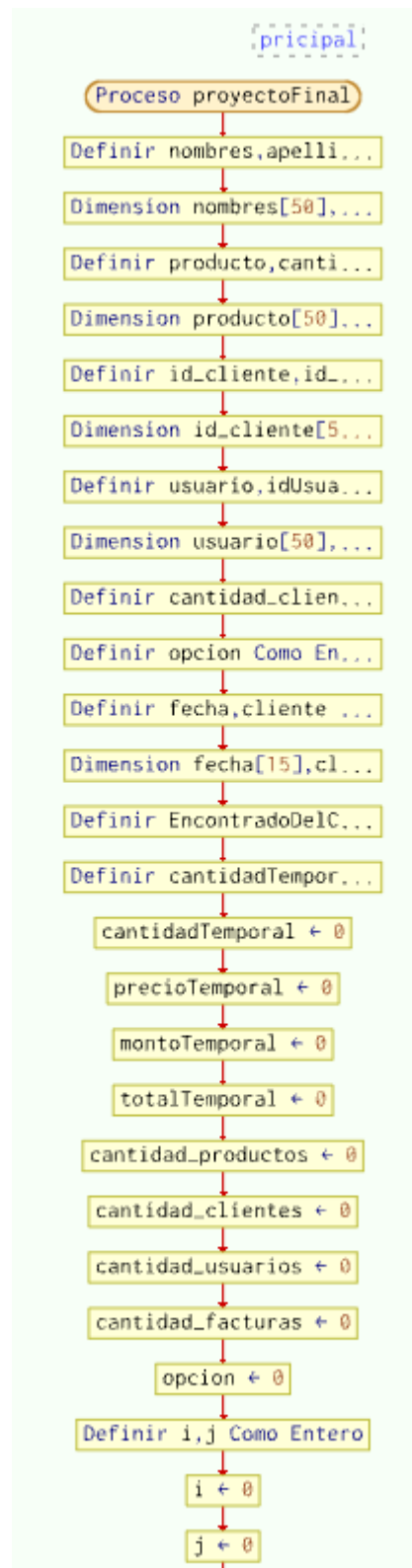
 Escribir "Opción inválida. Inténtalo de nuevo.";

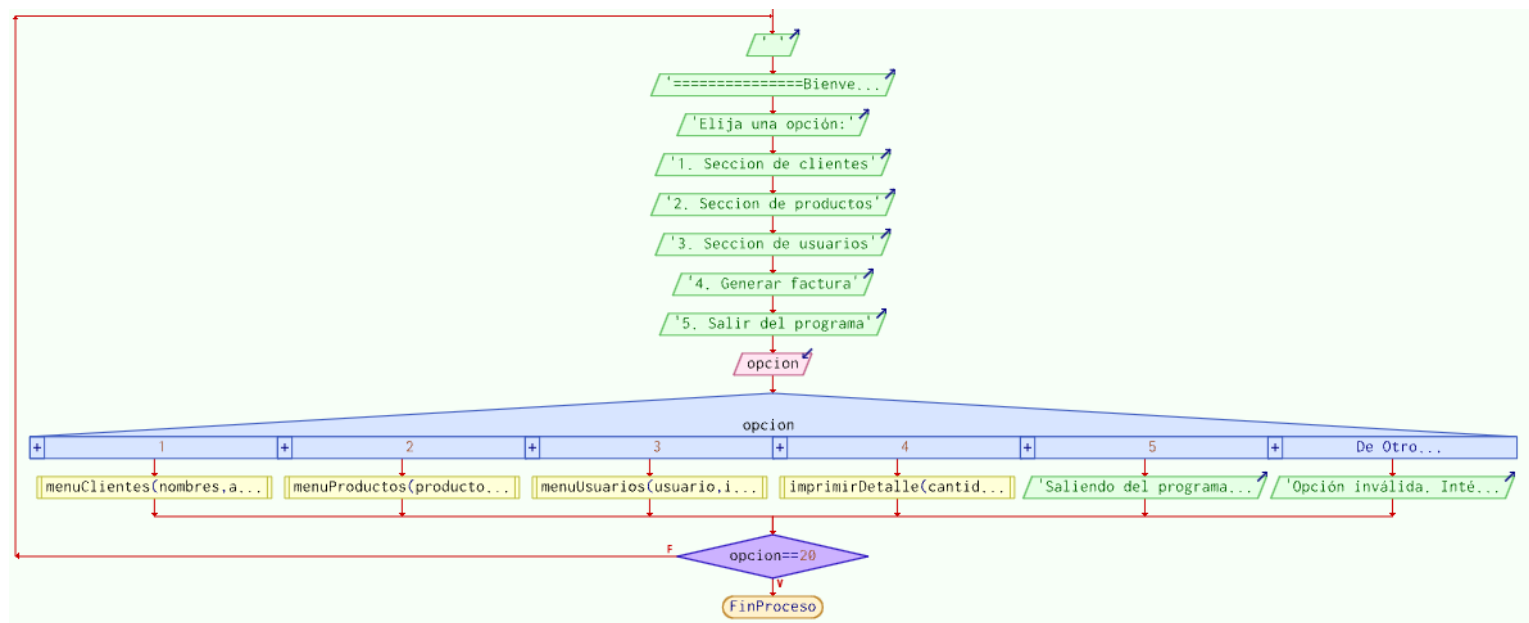
 FinSegun

 Hasta Que opcion == 20;

FinProceso

1.1.2. Diagrama de Flujo de Datos





Cientes

SubProceso agregarClientes(nombre...

'Ingrese el nombre del...

nombres[cantidad_clien...

'Ingrese el apellido d...

apellidos[cantidad_cli...

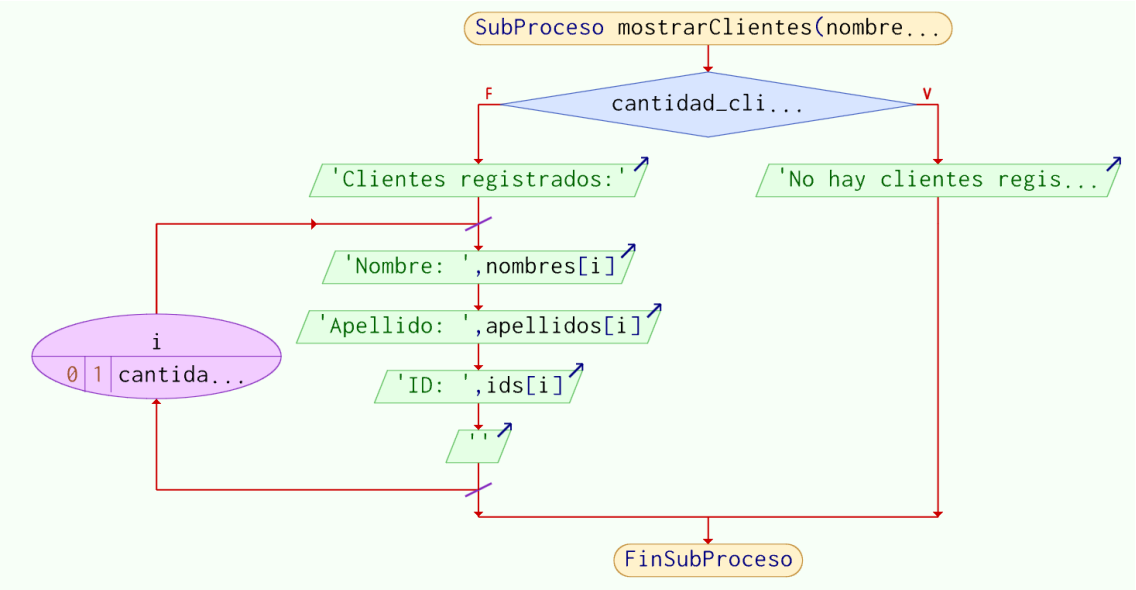
'Ingrese el ID del cli...

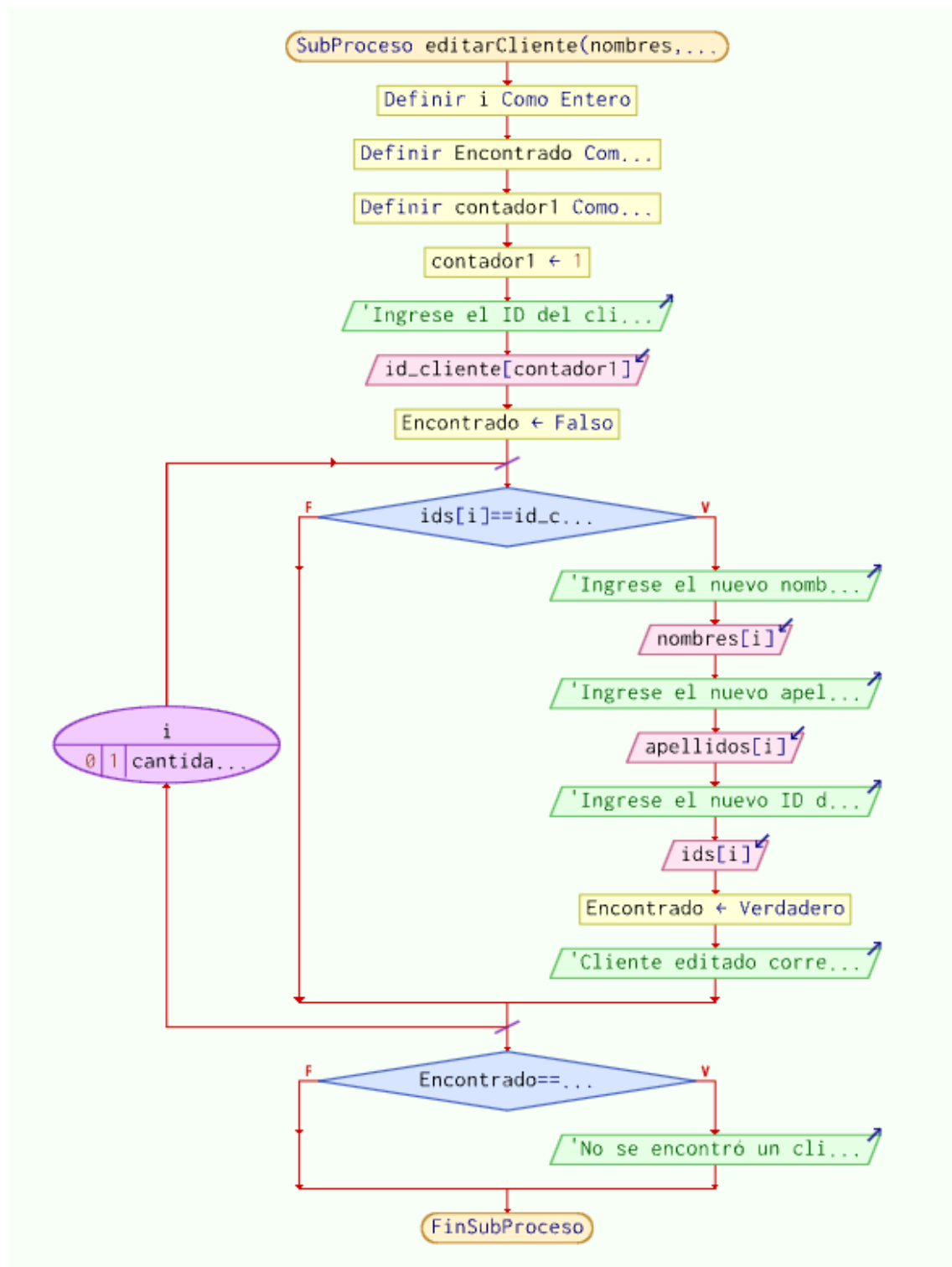
ids[cantidad_clientes]

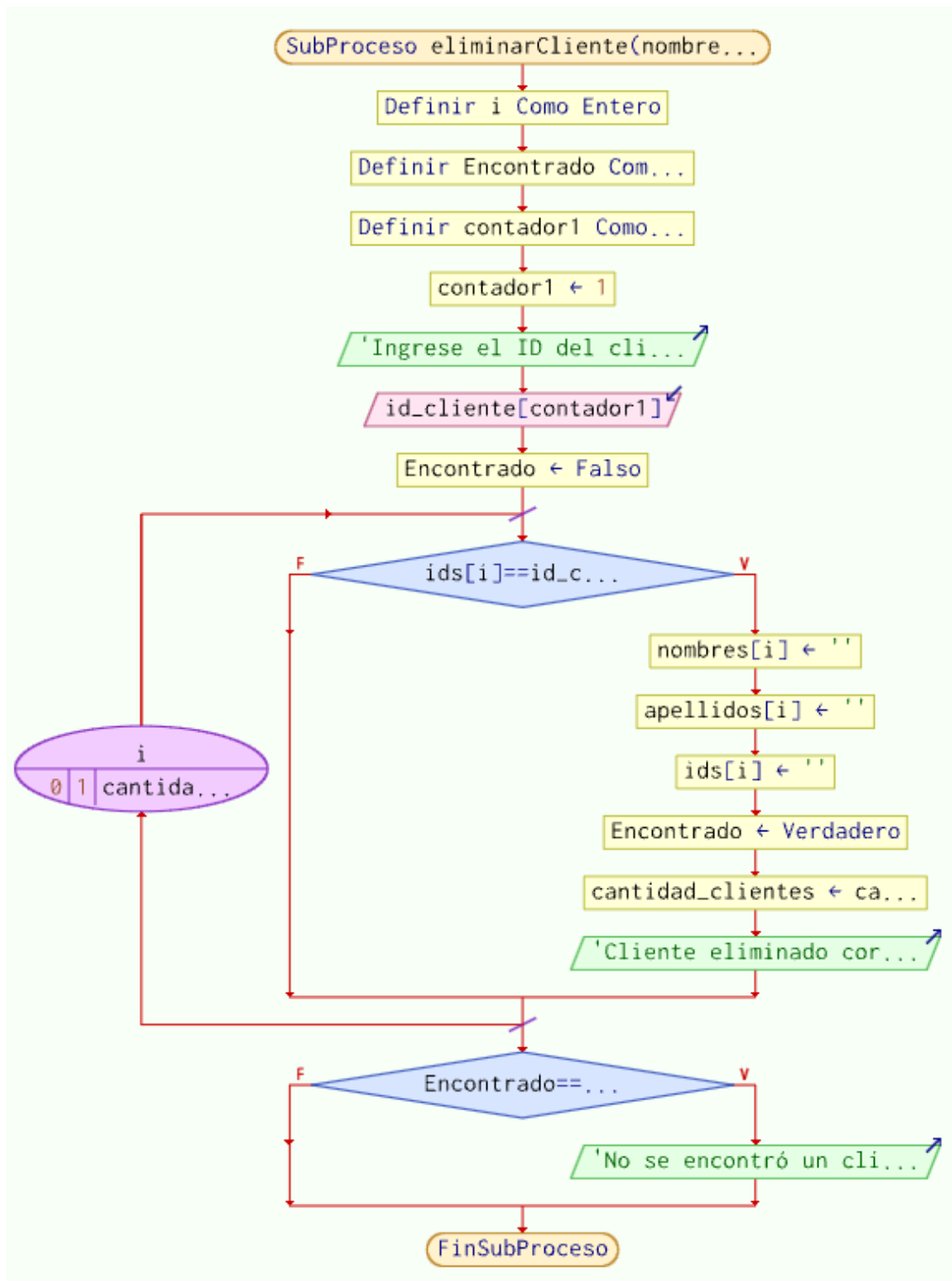
cantidad_clientes \leftarrow ca...

'Cliente agregado corr...

FinSubProceso







Productos

SubProceso agregarProducto(produc...

'Ingrese el nombre del...

producto[cantidad_prod...

'Ingrese la descripcio...

descripcion[cantidad_p...

'Ingrese el ID del pro...

idProducto[cantidad_pr...

'Ingrese la cantidad d...

cantidad[cantidad_prod...

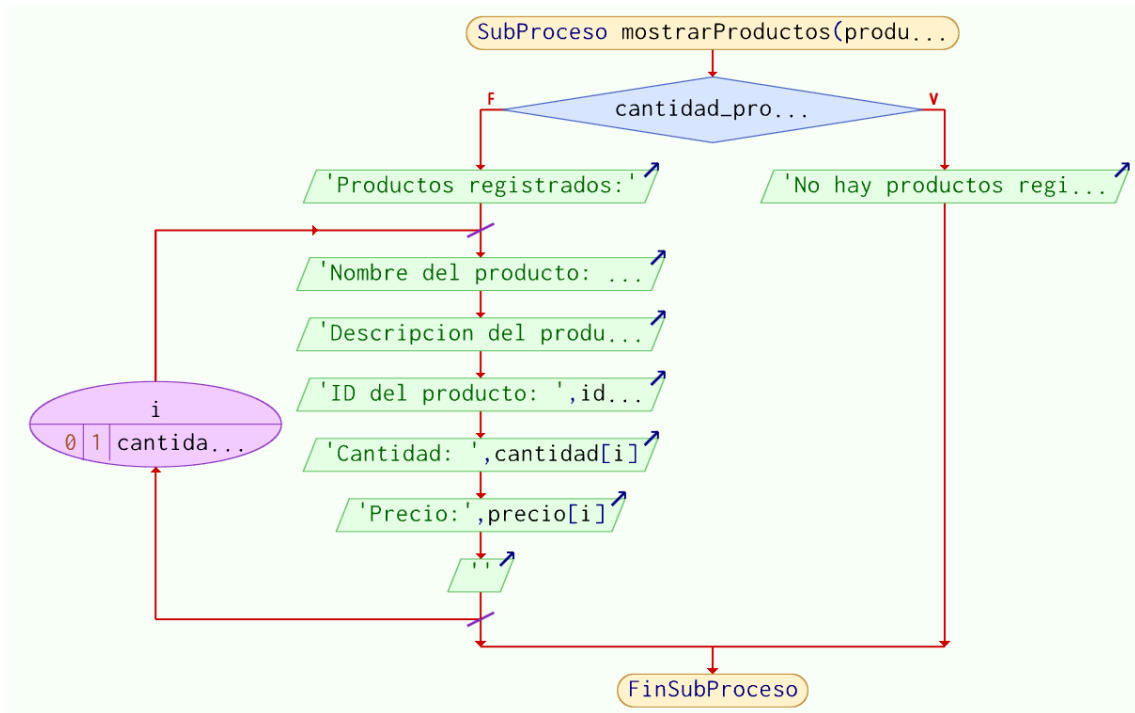
'Dime el precio del pr...

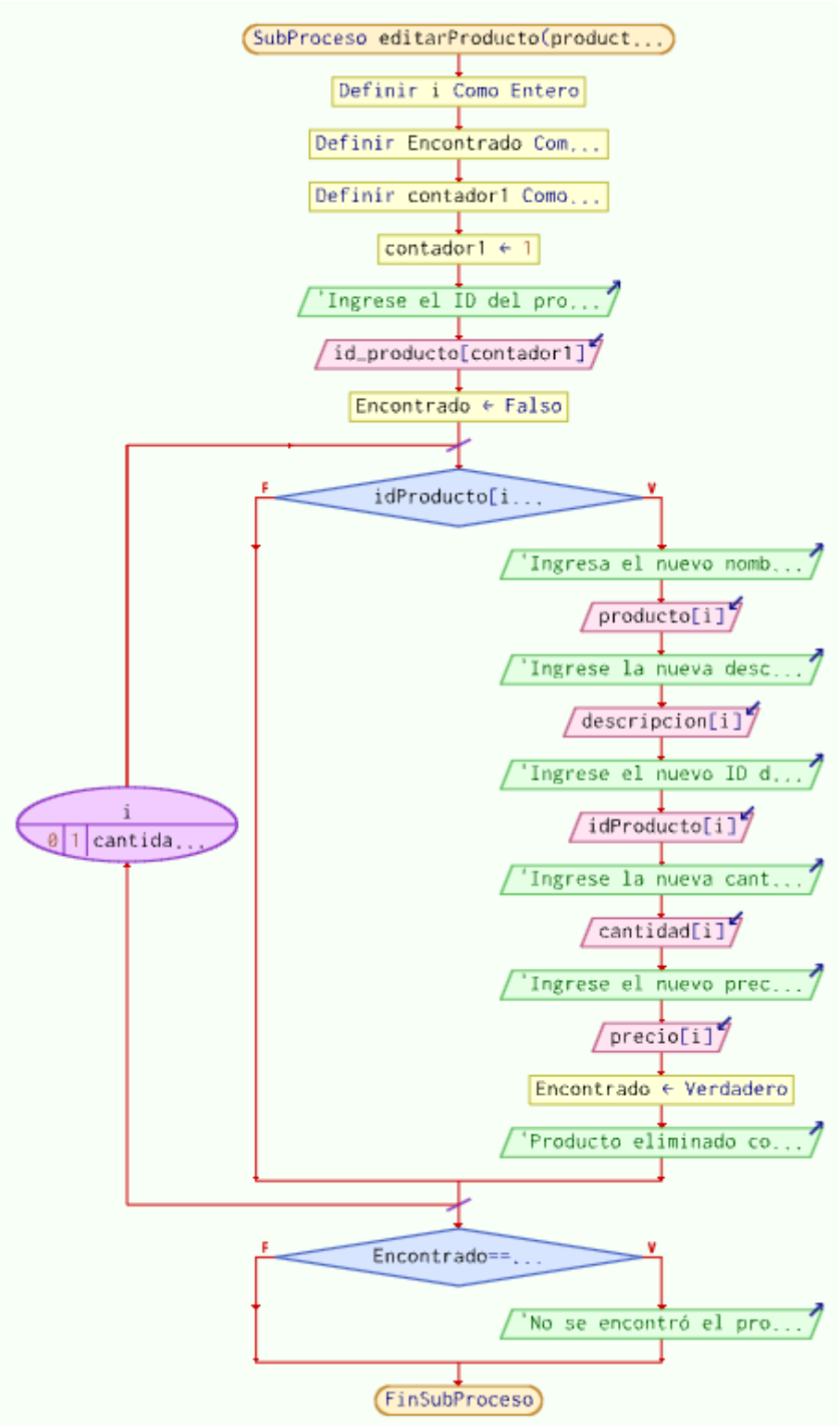
precio[cantidad_produc...

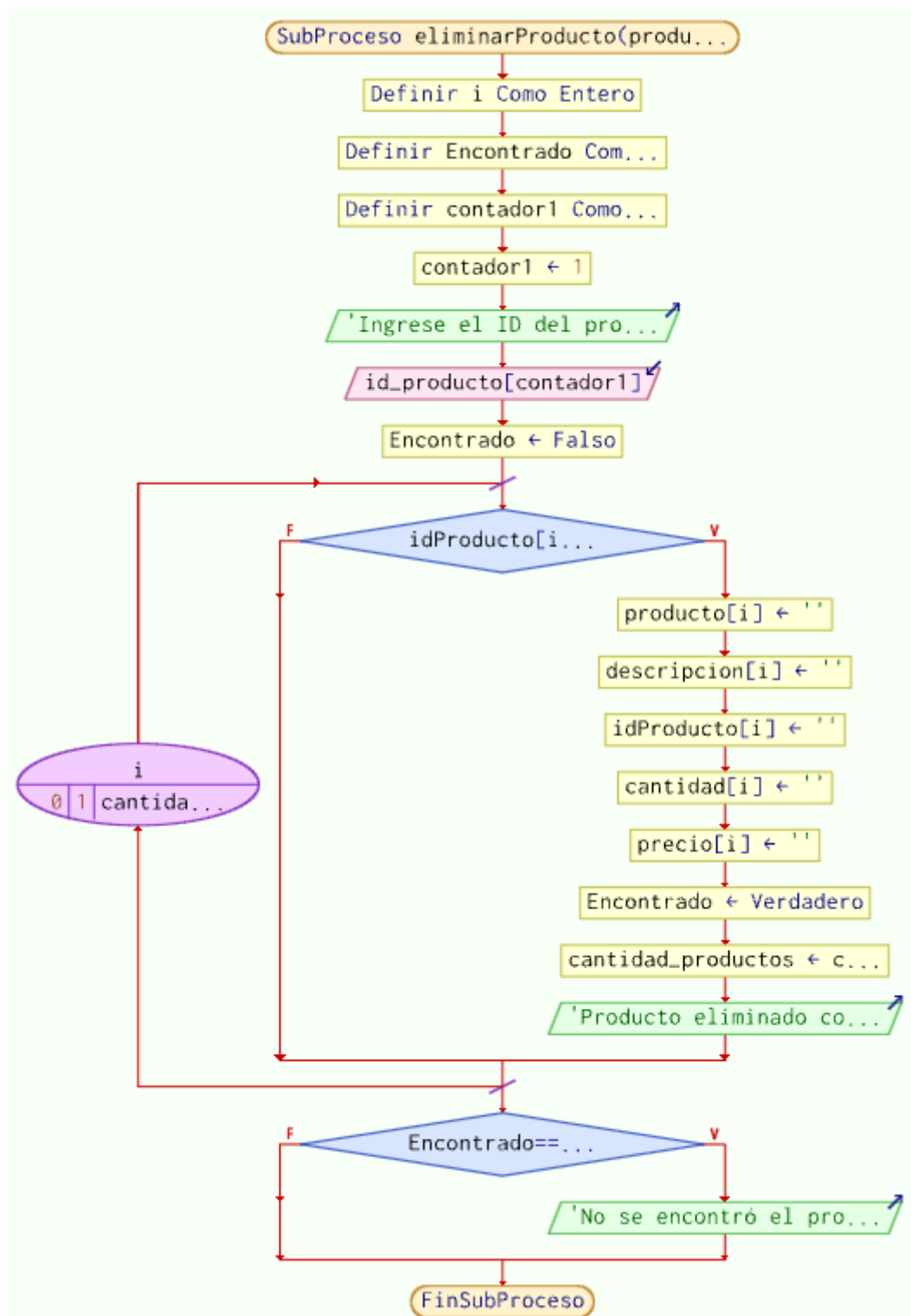
cantidad_productos \leftarrow c...

'Producto agregado cor...

FinSubProceso







usuarios

SubProceso agregarUsuario(usuario...

'Nombre del usuario: '

usuario[cantidad_usuar...

'Ingrese su ID: '

idUsuario[cantidad_usu...

'Ingrese su contraseña'

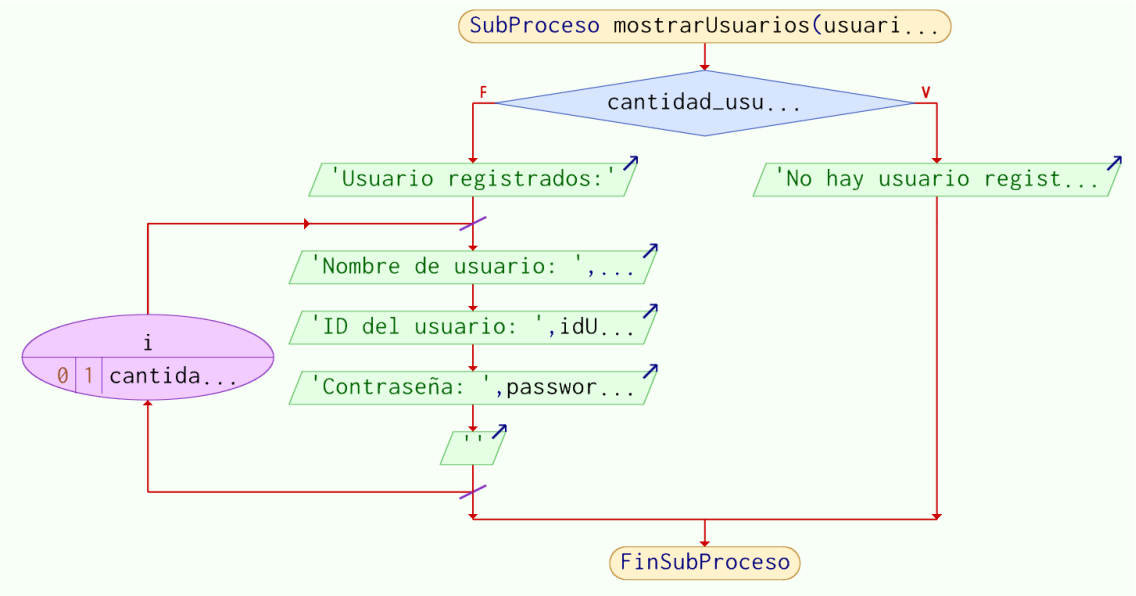
password[cantidad_usua...

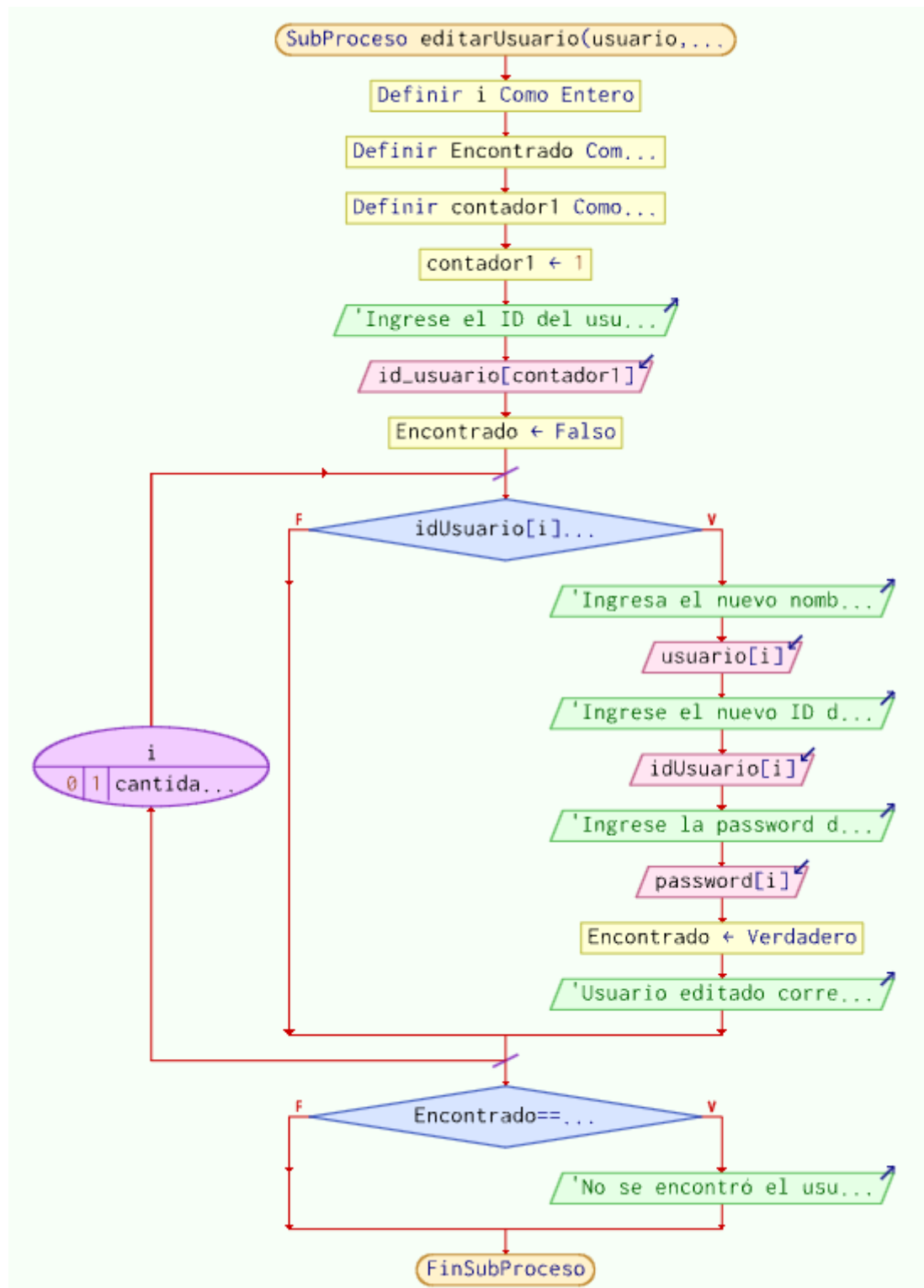
'Ingrese la cantidad d...

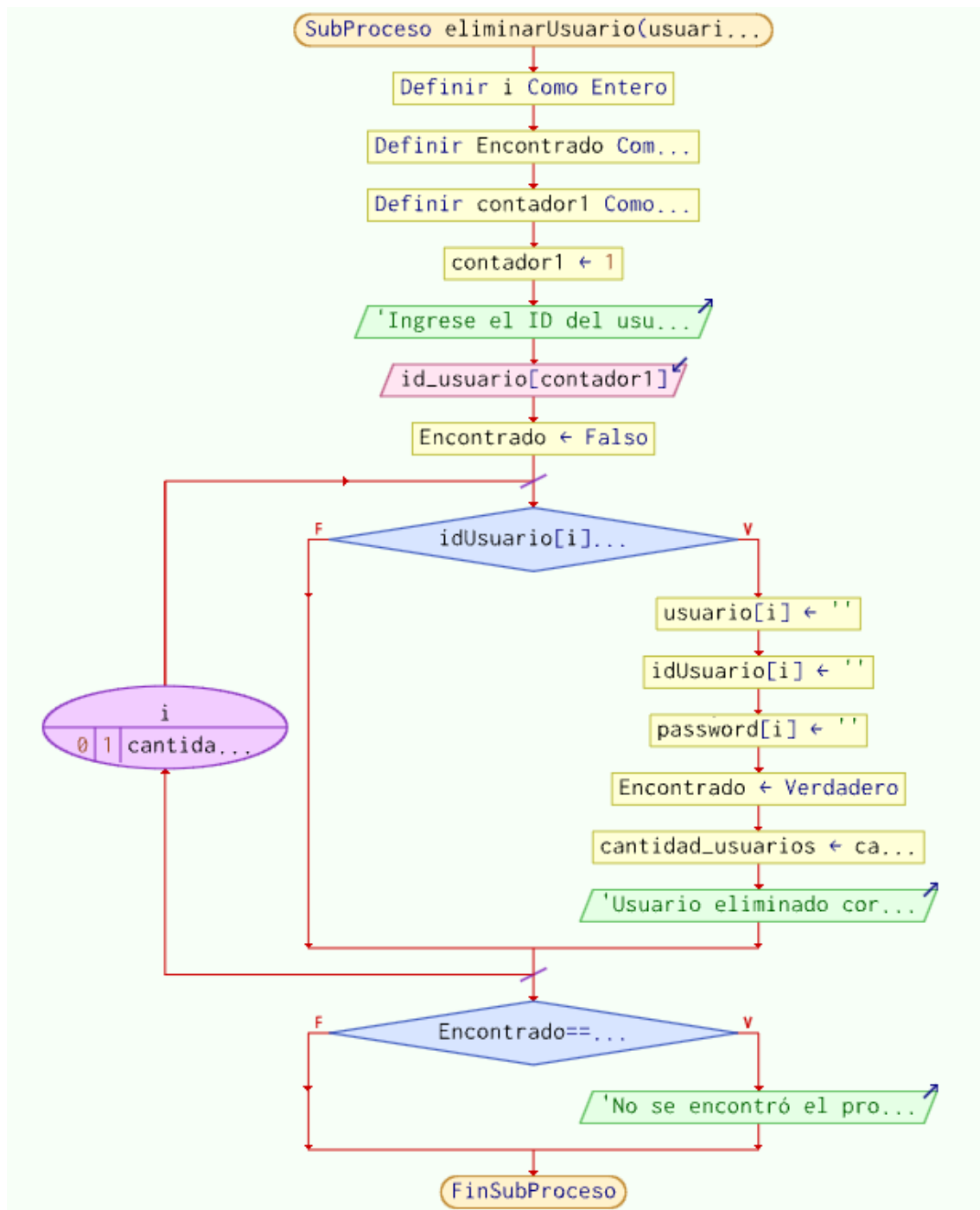
cantidad_usuarios \leftarrow ca...

'Usuario agregado corr...

FinSubProceso







facturas

SubProceso encabezadoFactura(cant...

'=====Bienve...

'Direccion: Masaya'

'Telefono: 25613023'

'Factura #: ', cantidad...

'Fecha: ', fecha[cantid...

'=====...

'Cliente: ', cliente[ca...

FinSubProceso

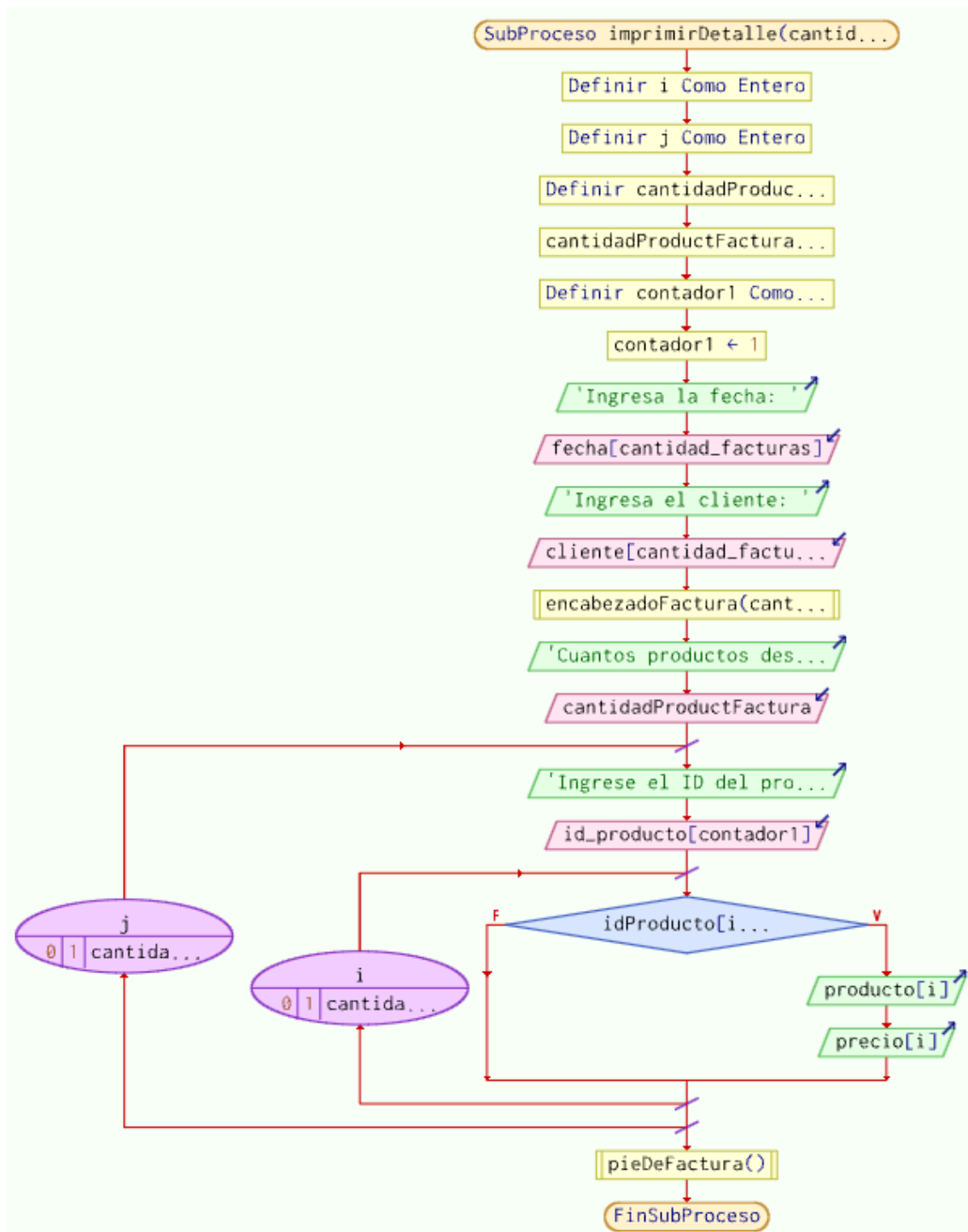
SubProceso pieDeFactura()

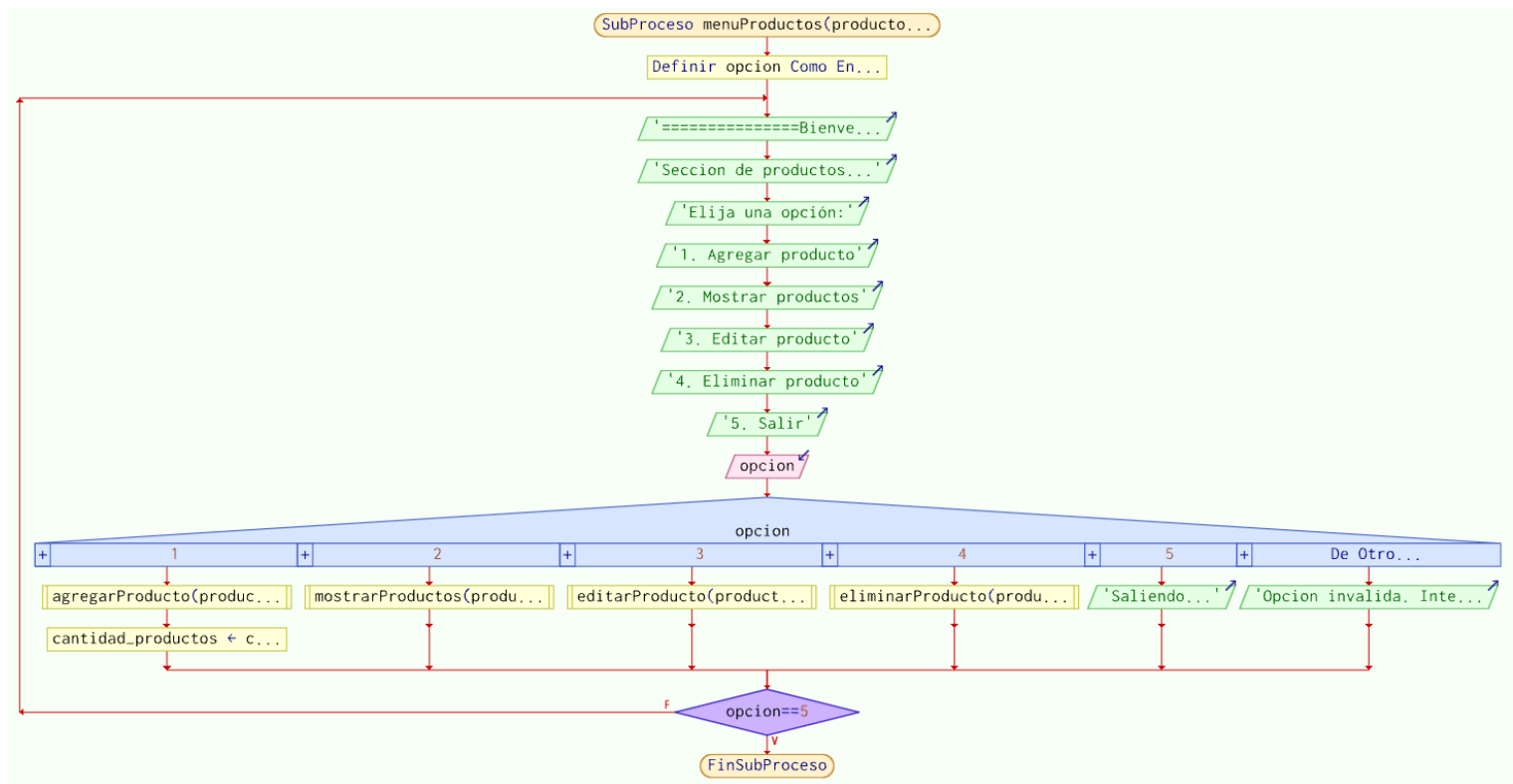
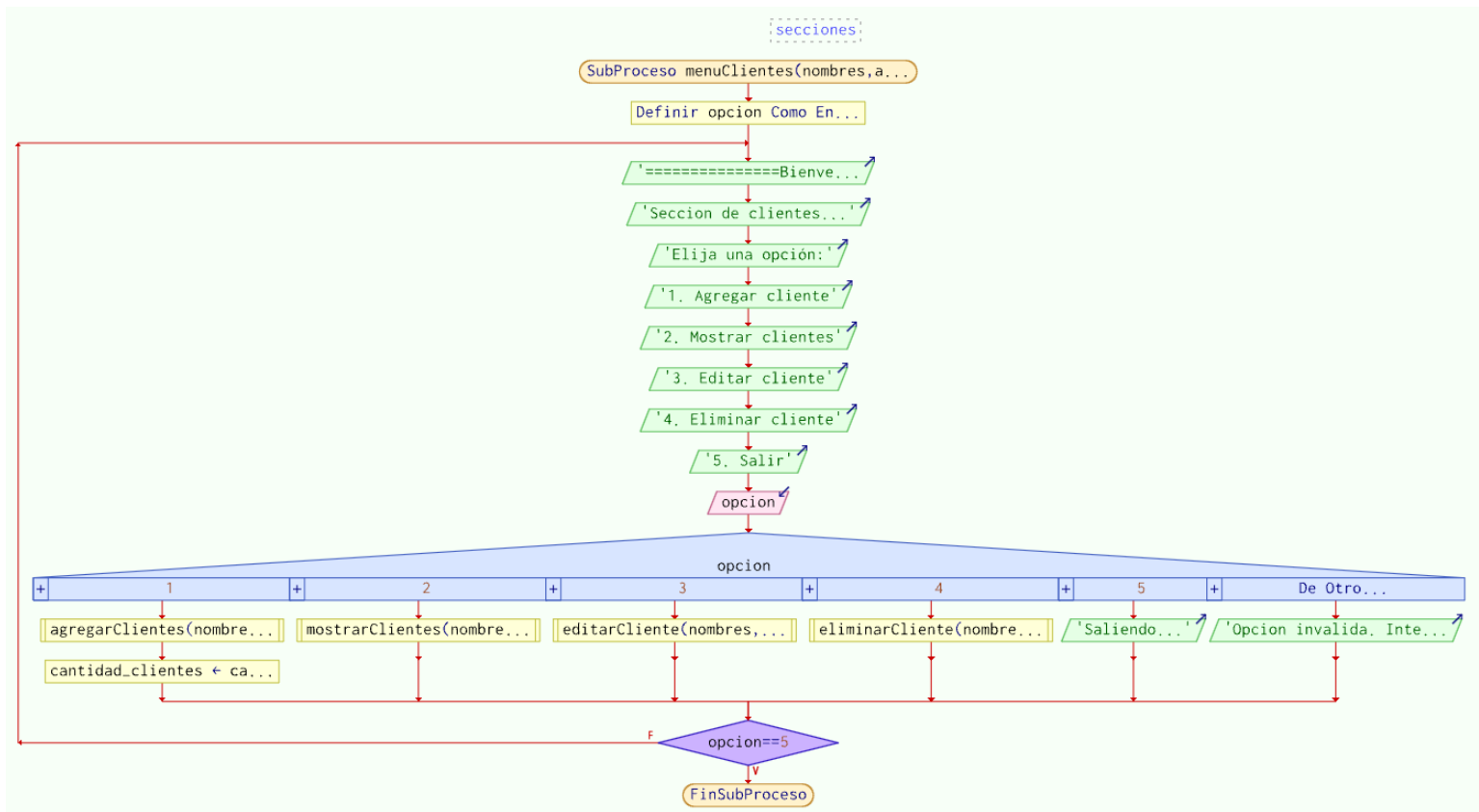
'===== ...'

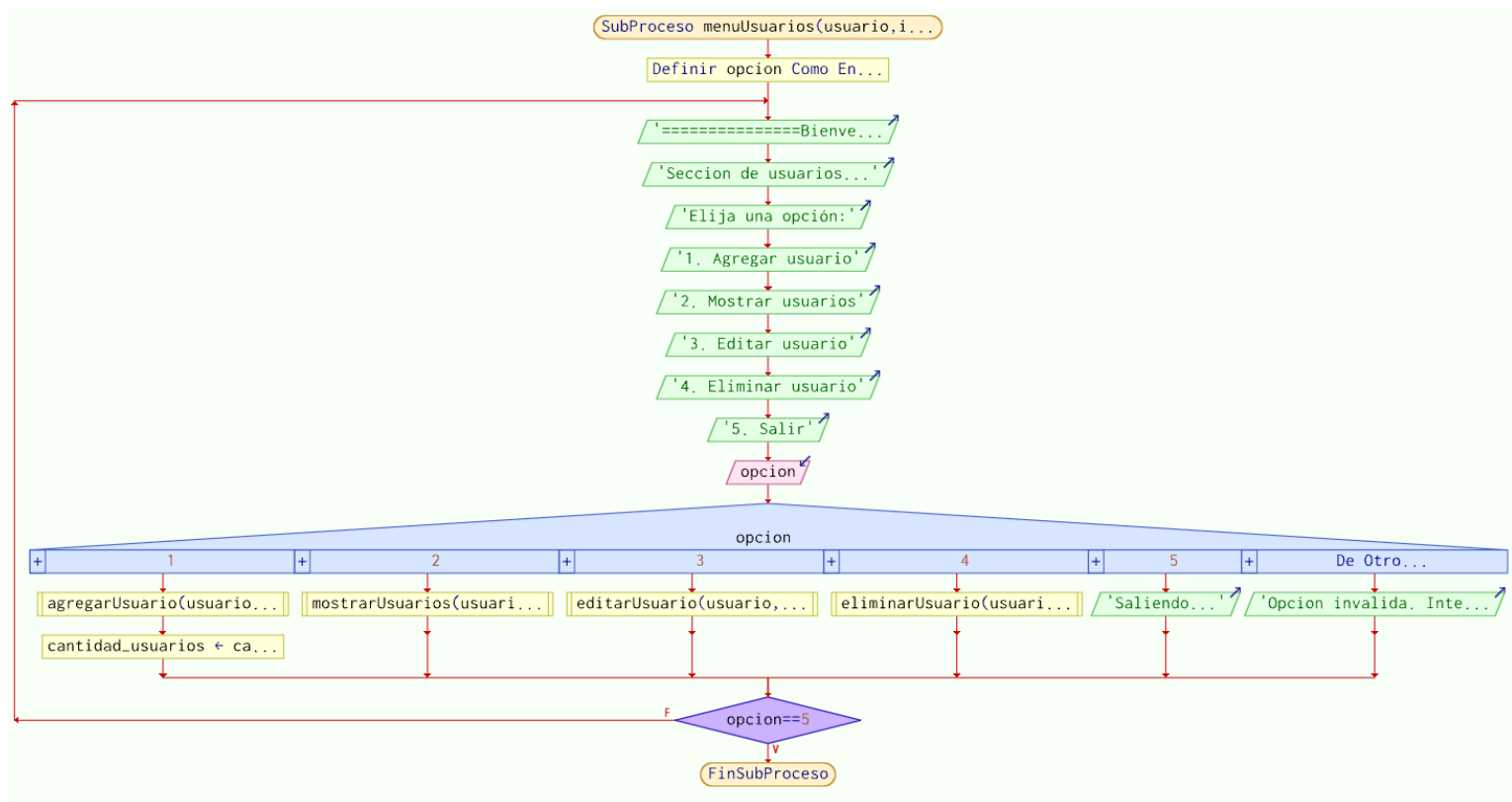
'Gracias por su compra'

'===== ...'

FinSubProceso







3. FASE DE CODIFICACIÓN Y DEPURACIÓN

3.1. Estructuras de datos del proyecto

Estructura de clientes

```
include <iostream>
#include <stdio.h>
#include <string.h>
#define MAX 200

typedef struct
{
    char ID[14];
    char nombre[30];
    char apellido[30];
}cliente;

cliente Clien[MAX];
```

Estructura de usuarios

```
#include <iostream>
#include <stdio.h>
#include <string.h>
#define MAX 200

typedef struct
{
    char IDU[14];
    char nombreU[30];
    char email[100];
    char password[20];
} usuario;

usuario usuarios[MAX];
producto productos[MAX];
```


Estructura de productos

```
#include <iostream>
#include <stdio.h>
#include <string.h>
#define MAX 200
```

```
typedef struct
{
    char IDP[10];
    char nombreP[30];
    char descripcion[100];
    float precio;
}producto;
```

Estructura de facturas

```
#include <iostream>
#include <stdio.h>
#include <string.h>
#define MAX 200
```

```
typedef struct
{
    char IDP[10];
    char nombreP[30];
    char descripcion[100];
    float precio;
}producto;
```

Auxiliar.h

```
#ifdef _WIN32 // Para Windows
#include <windows.h>
void gotoxy4(int x, int y) {
    COORD coord = { x, y };
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
#elif __linux__ // Para Linux
#include <ncurses.h>
void gotoxy(int x, int y) {
    move(y, x);
}
#else // Para otros sistemas operativos
void gotoxy(int x, int y) {
    printf("\033[%d;%df", y, x);
}
#endif
```

3.1. Código fuente y salidas

MENU PRINCIPAL

```
-----Bienvenido a MCSerigraph-----
Elegir el tipo de cuenta
1. Soy Administrador
2. Soy Empleado
3. Soy Cliente
4. Salir
_
```

MENU ADMIN CON TODAS LAS FUNCIONES

```
-----Bienvenido al menu de Administrador de MCSerigraph-----
Seleccione una de las opciones:
1. Seccion de clientes
2. Seccion de productos
3. Seccion de usuarios
4. Seccion de facturas
5. Salir
```

MENU CLIENTES

```
-----Bienvenido al menu de MCSerigraph-----
C Seleccione una de las opciones
a Ver la cantidad de clientes registrados 1
1. Agregar cliente
2. Editar cliente
3. Eliminar cliente
4. Buscar cliente
5. Mostrar todos los clientes
6. Salir
  Digite la opcion
```

MENU PRODUCTOS

```
-----Bienvenido al menu de MCSerigraph-----
C Seleccione una de las opciones
f Cantidad de productos almacenados 2
1. Agregar producto
2. Editar producto
3. Eliminar producto
4. Buscar producto
5. Mostrar todos los productos
6. Salir
  Digite la opcion
```

MENU USUARIOS

```
-----Bienvenido al menu de MCSerigraph-----
C Seleccione una de las opciones
f Ver la cantidad de clientes registrados 0
1. Agregar usuario
2. Editar usuario
3. Eliminar usuario
4. Buscar usuario
5. Mostrar todos los usuario
6. Salir
  Digite la opcion
```

MENU FACTURAS

```
-----Bienvenido al menu de MCSerigraph-----
C Seleccione una de las opciones
f Ver la cantidad de facturas hasta el momento 3
1. Agregar datos a la factura
4. Salir
  Digite la opcion
```