

**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](#_Toc134430749)

[1.1. PLANTEAMIENTO DEL PROBLEMA 1](#_Toc134430750)

[1.2. DIAGRAMA DE ESTRUCTURA 1](#_Toc134430751)

[1.3. RESULTADOS DE LA FASE DE ANÁLISIS 1](#_Toc134430752)

[**2.** **FASE DE DISEÑO** 2](#_Toc134430753)

[1.1. ALGORITMOS 2](#_Toc134430754)

[1.1.1. Pseudocódigo 2](#_Toc134430755)

[1.1.2. Diagrama de Flujo de Datos 3](#_Toc134430756)

[**3.** **FASE DE CODIFICACIÓN Y DEPURACIÓN** 3](#_Toc134430757)

[3.1. Estructuras de datos del proyecto 3](#_Toc134430758)

[3.2. Código fuente y salidas 3](#_Toc134430759)

# **FASE DE ANÁLISIS**

## PLANTEAMIENTO DEL PROBLEMA

La empresa de Serigrafia MCSerigraph, nos contrató para crearles un sistema de registro adecuado para sus clientes, empleados y facturas. Asimismo, para las personas que gusten comprar los productos, tener una mayor facilidad para ver los productos y realizar el proceso de facturación

## DIAGRAMA DE ESTRUCTURA

## 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 especifico |
| 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 especifico 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 especifico |
| 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 especifico 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 especifico |
| 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 especifico 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** |
|  |
| **Agregando clientes…**  **ID del cliente:**  **Nombre del cliente:**  **Apellido del cliente:**  **Agregando productos…**  **ID del producto:**  **Nombre del producto:**  **Descripción del producto:**  **Precio del producto:**  **Agregando usuarios…**  **ID de usuario:**  **Nombre de usuario:**  **Email de usuario:**  **Contraseña del usuario:**  **Generando factura…** |

# **FASE DE DISEÑO**

## ALGORITMOS

## 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

Hasta Que opcion == 5;

FinSubProceso

**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:

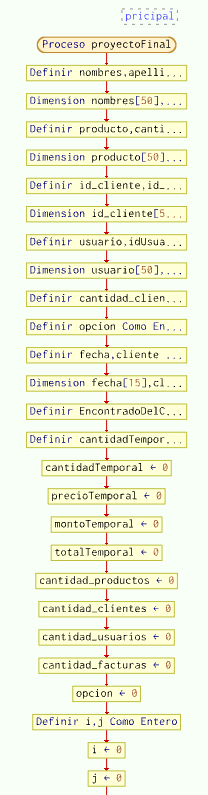
Escribir "Opción inválida. Inténtalo de nuevo.";

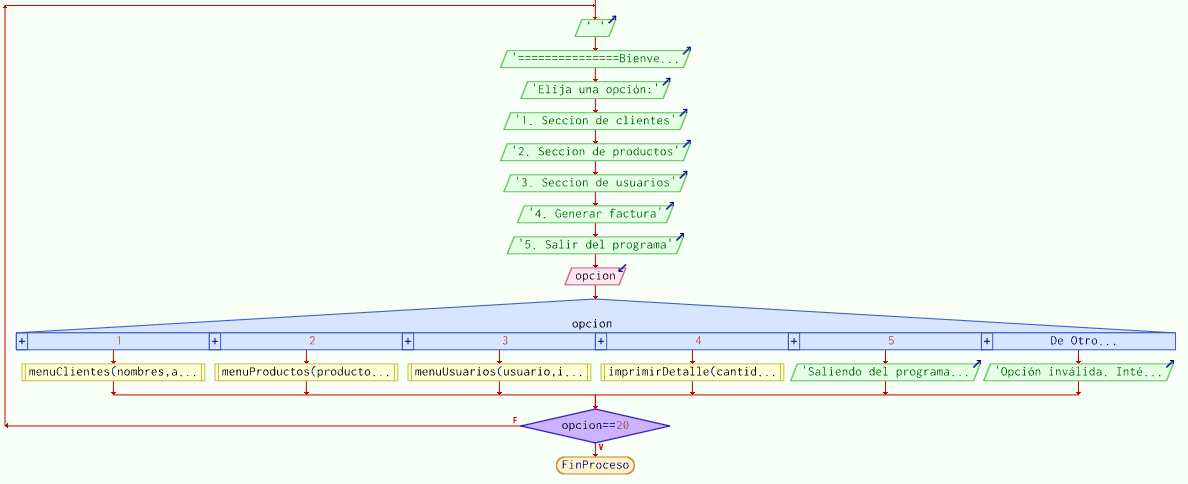
FinSegun

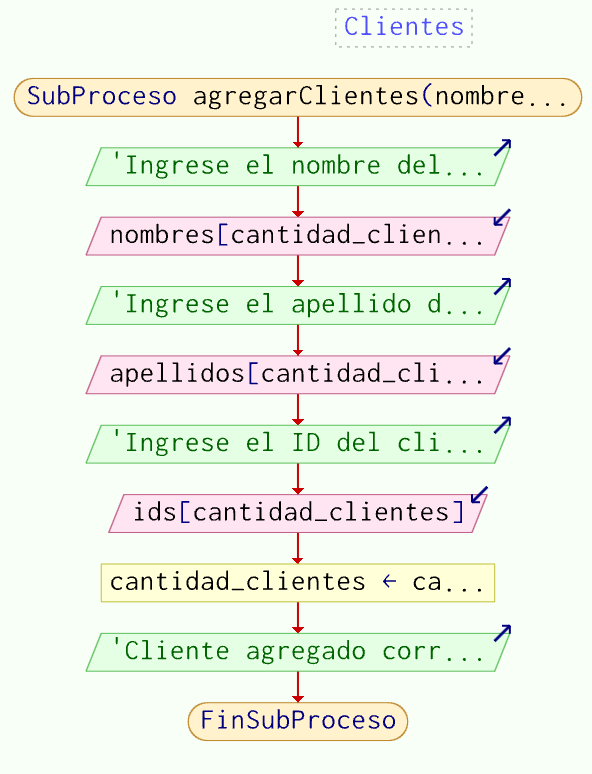
Hasta Que opcion == 20;

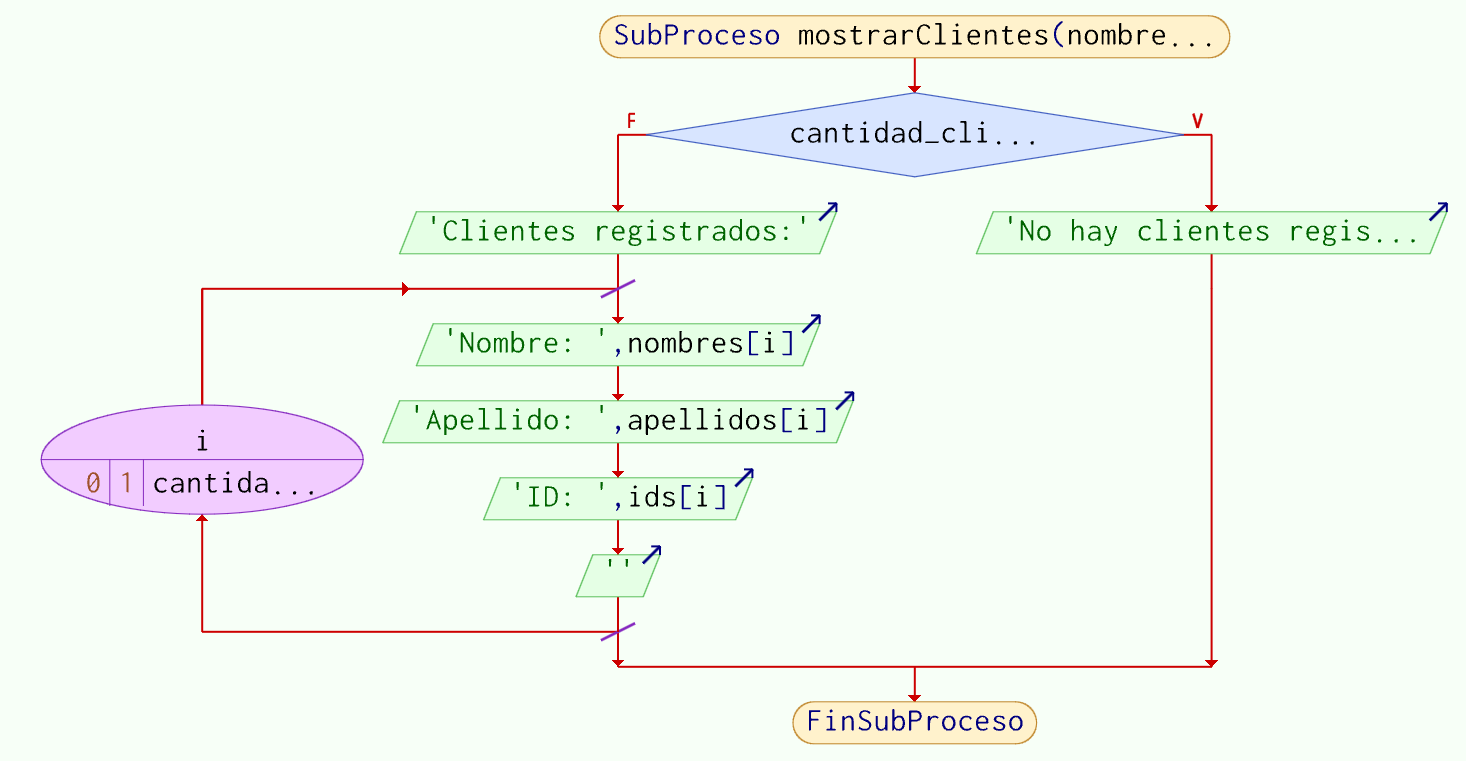
FinProceso

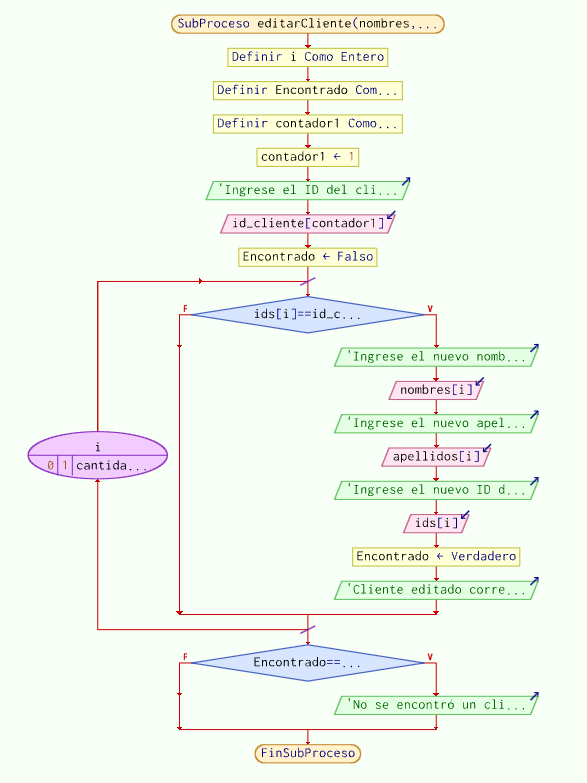
## Diagrama de Flujo de Datos

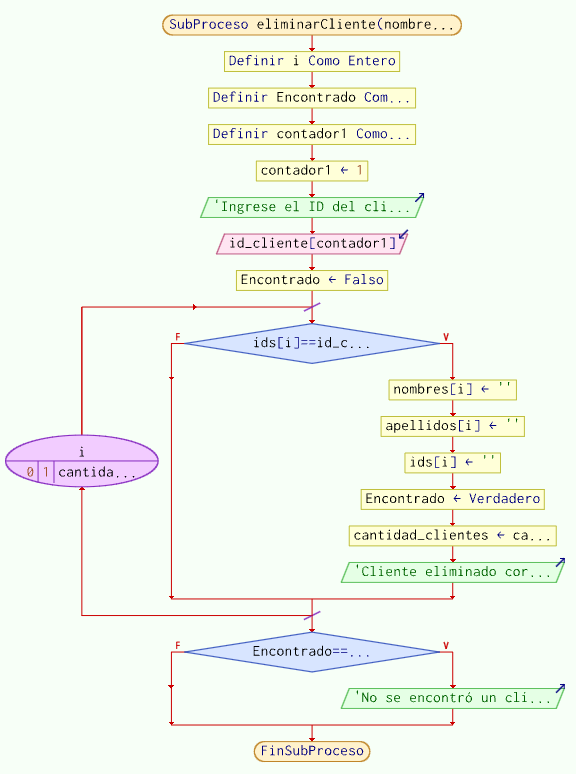




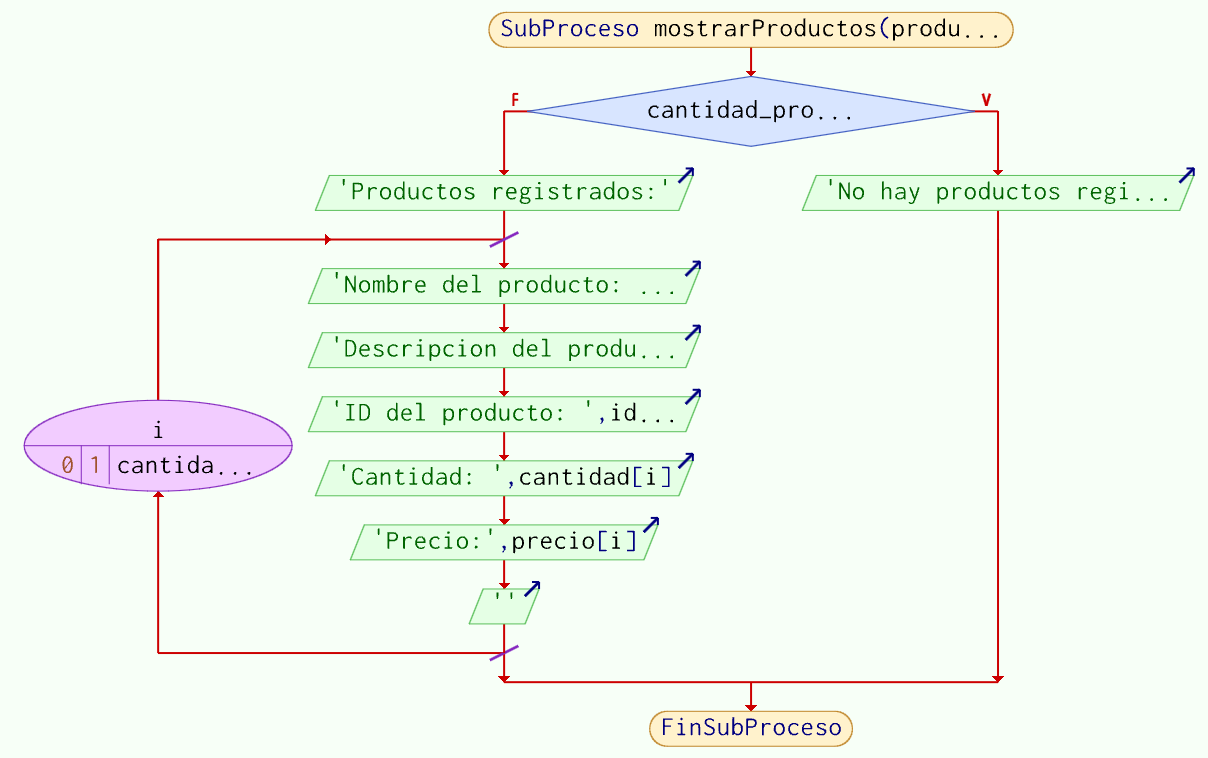


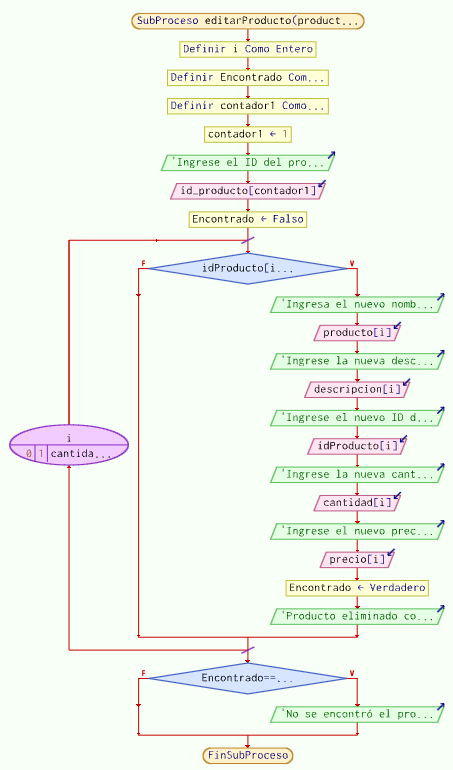


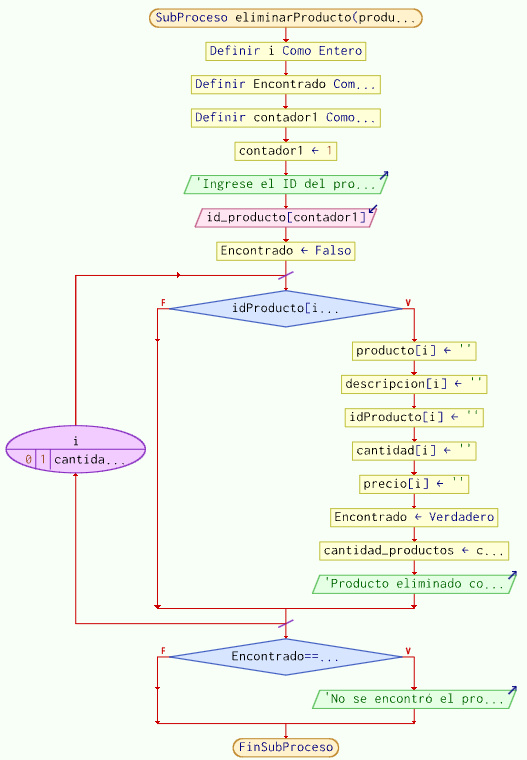


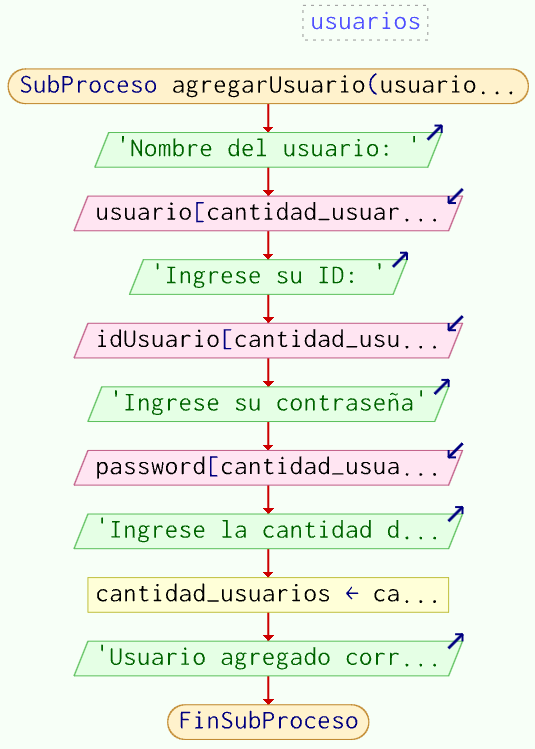


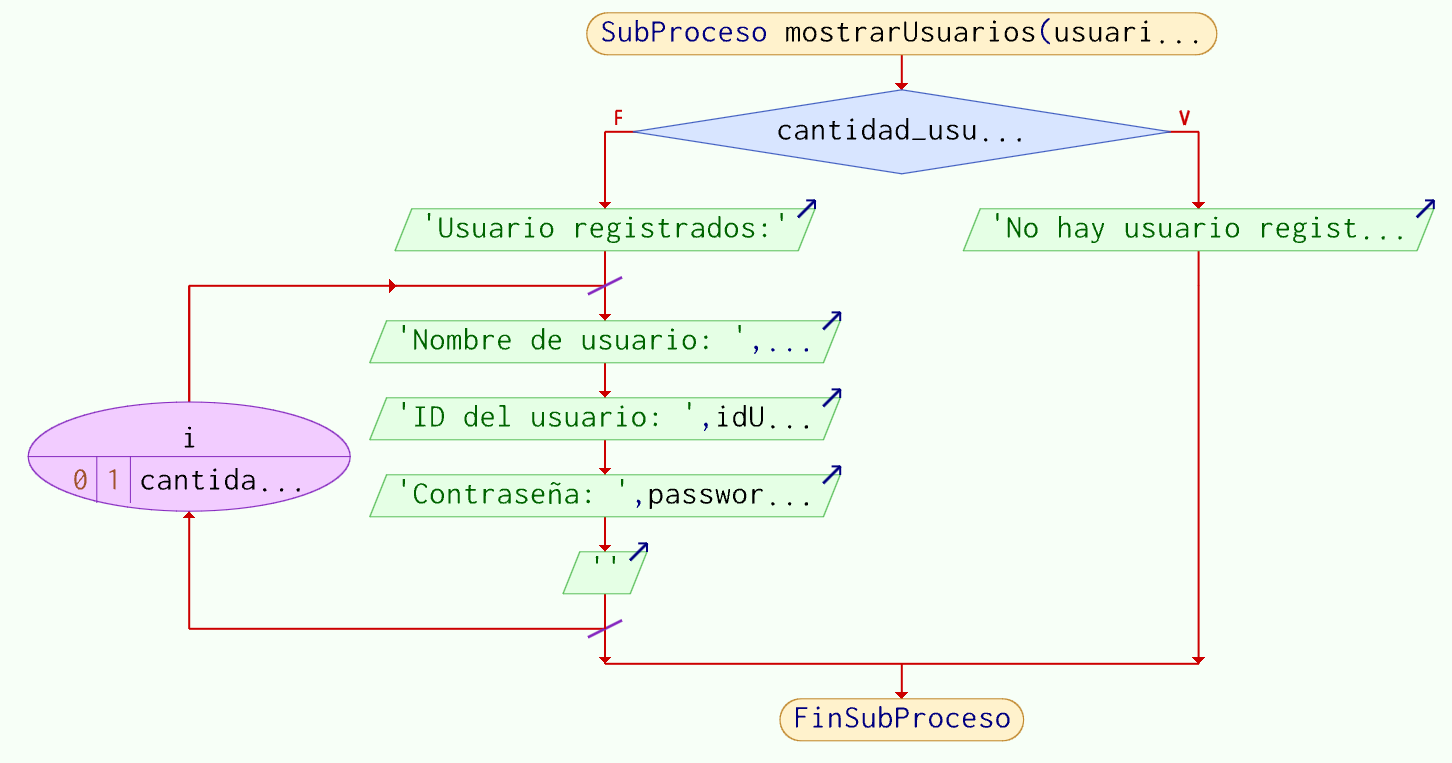


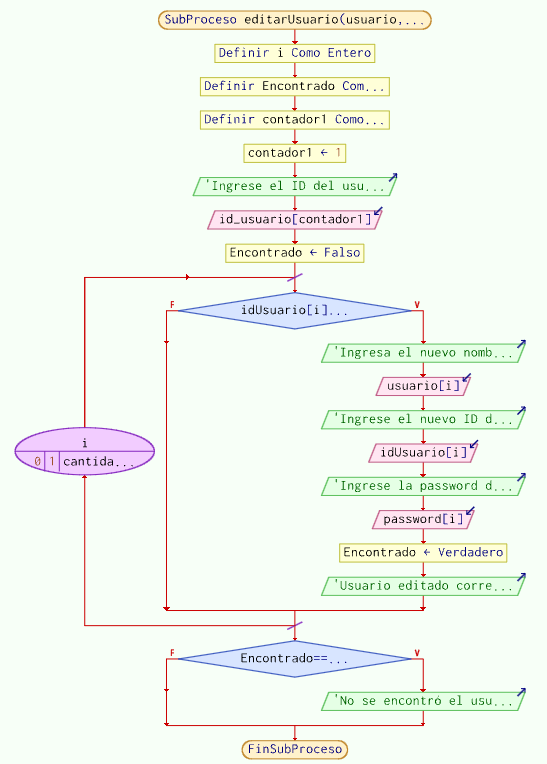


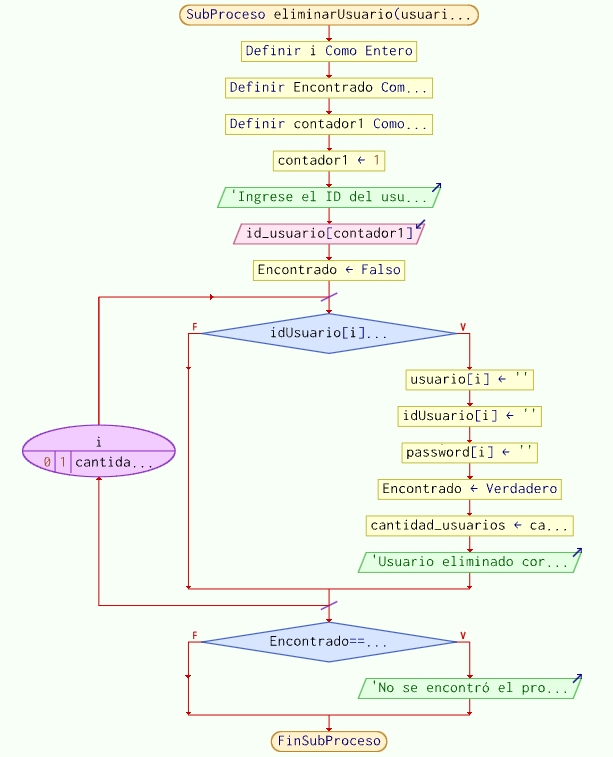


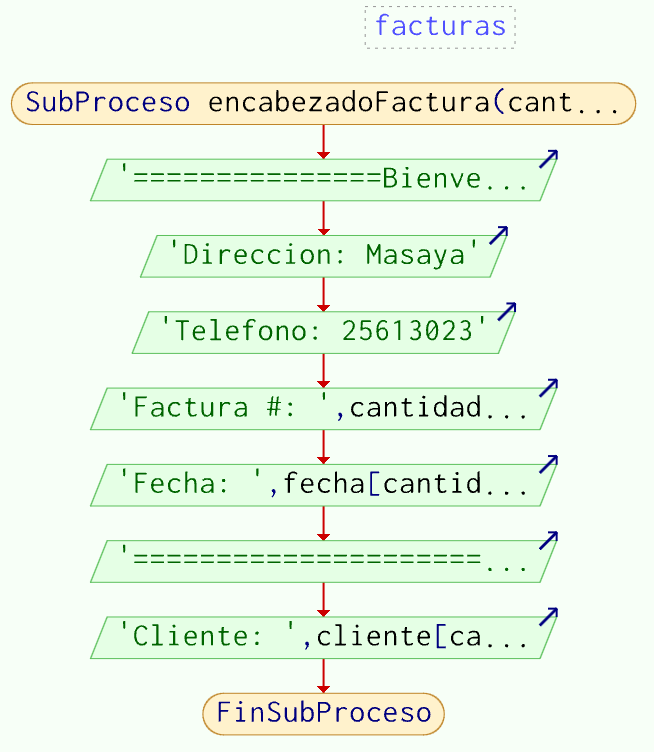


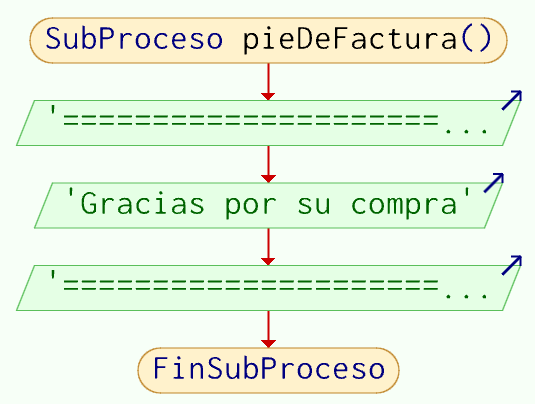


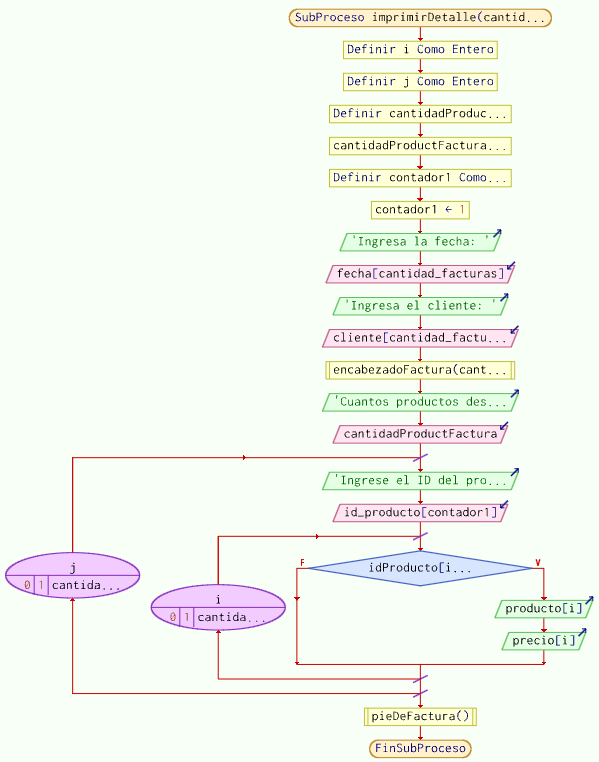


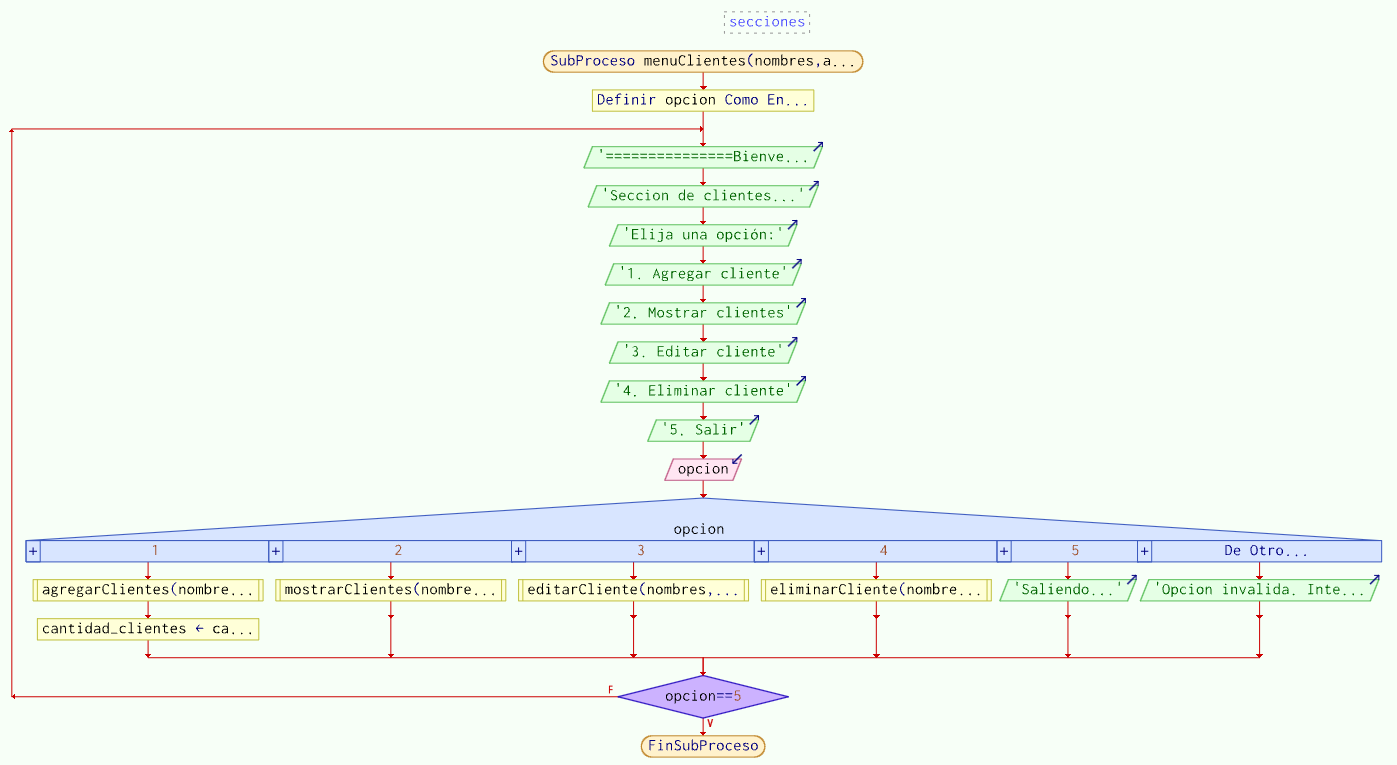
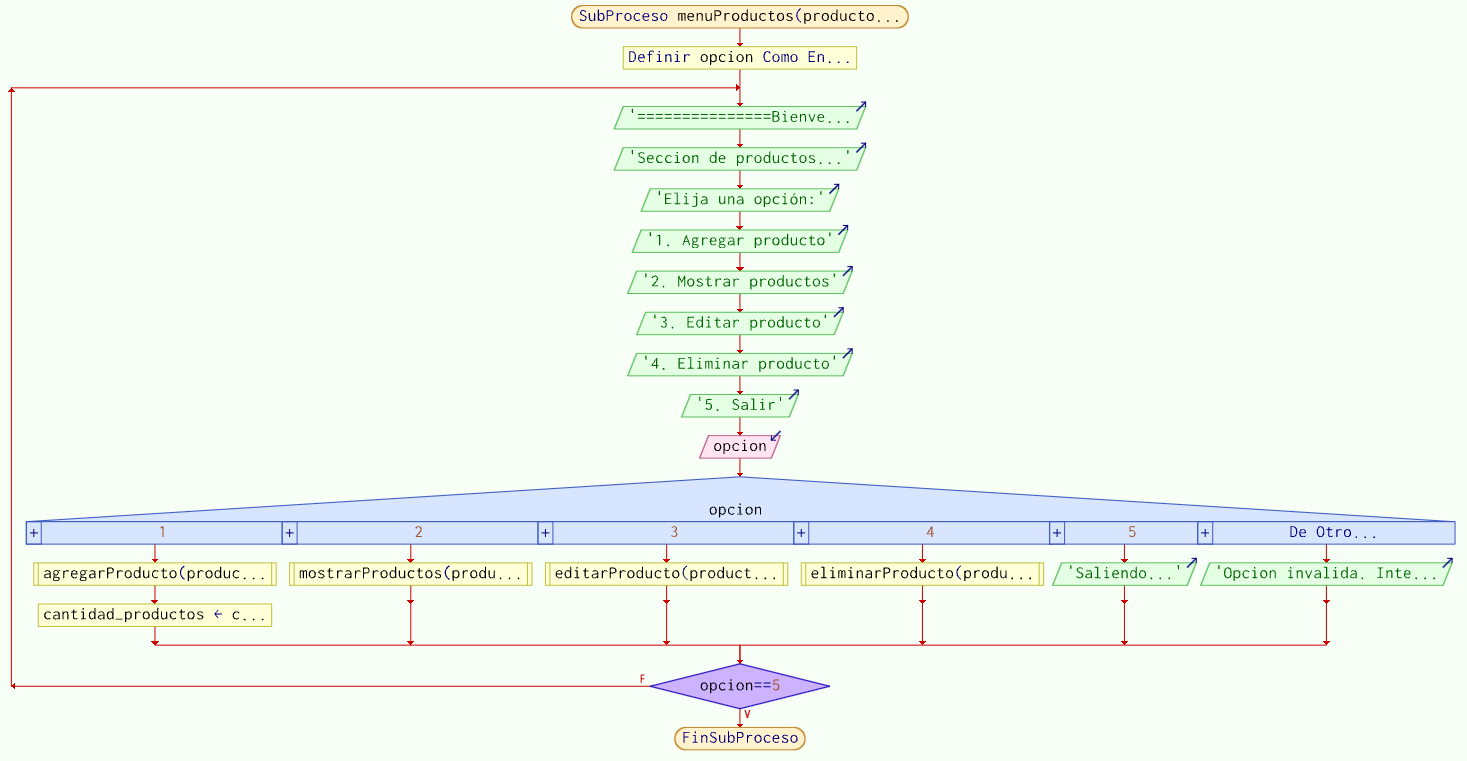


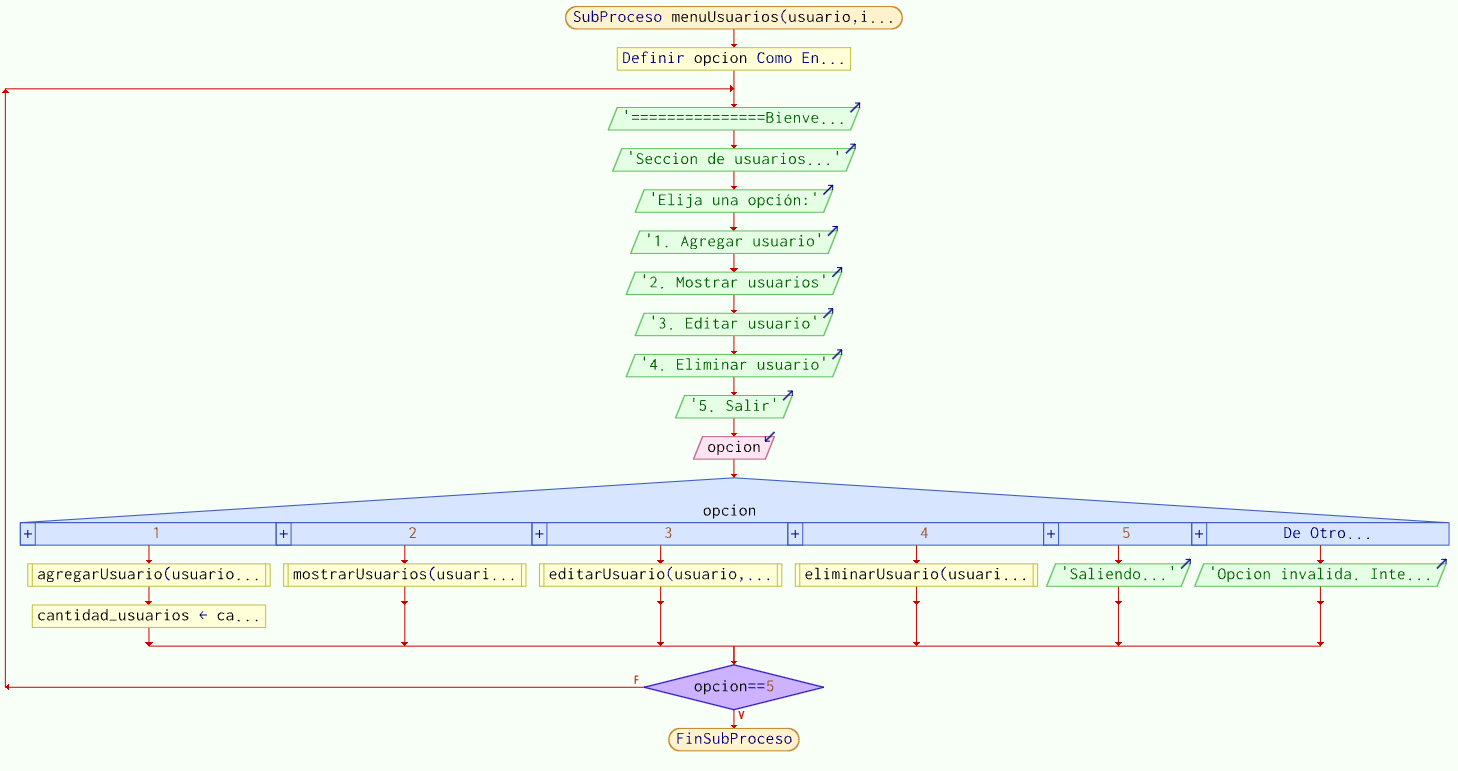












# **FASE DE CODIFICACIÓN Y DEPURACIÓN**

## 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 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;

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;

producto productos[MAX];

**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);

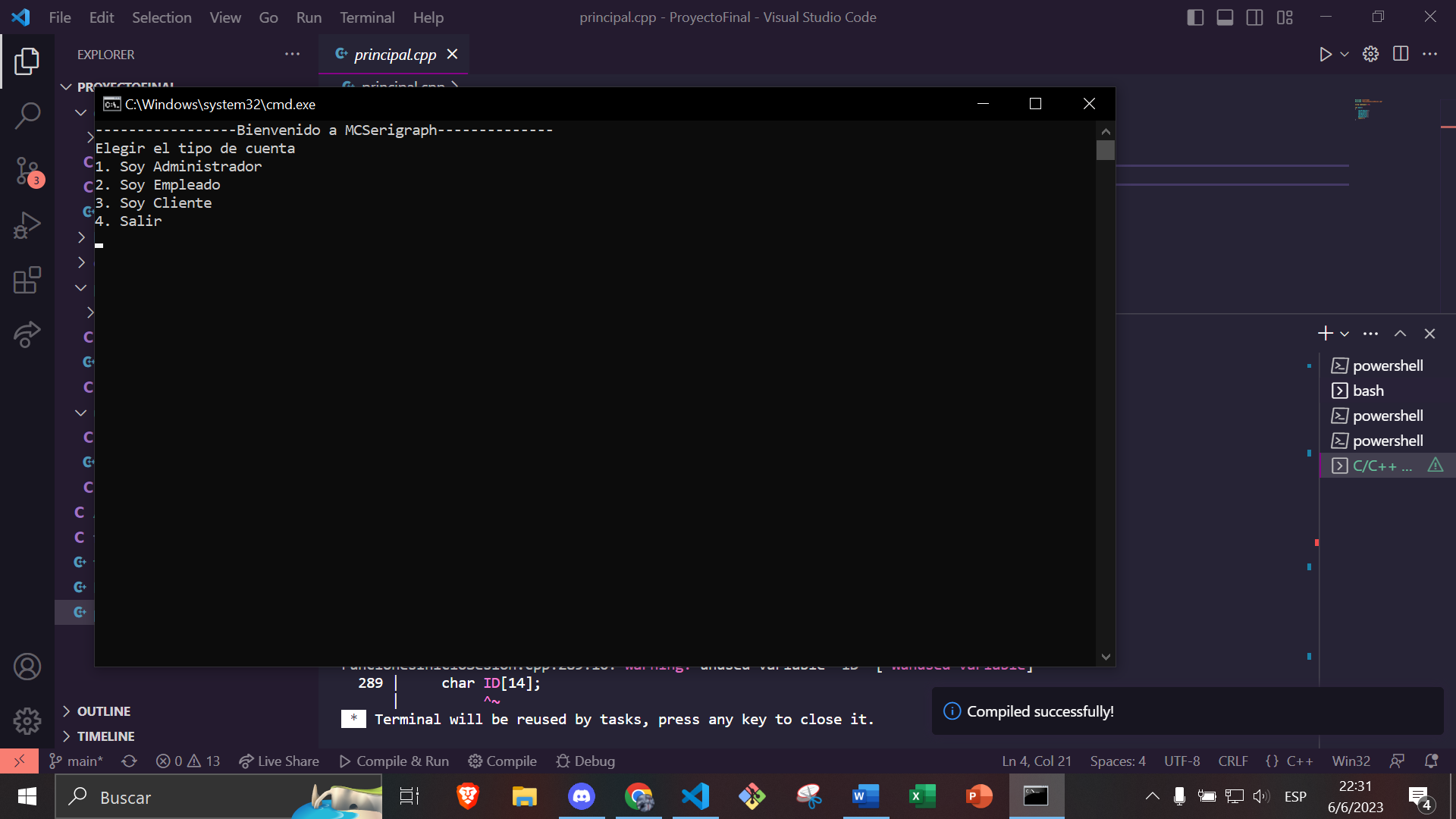
}

#endif

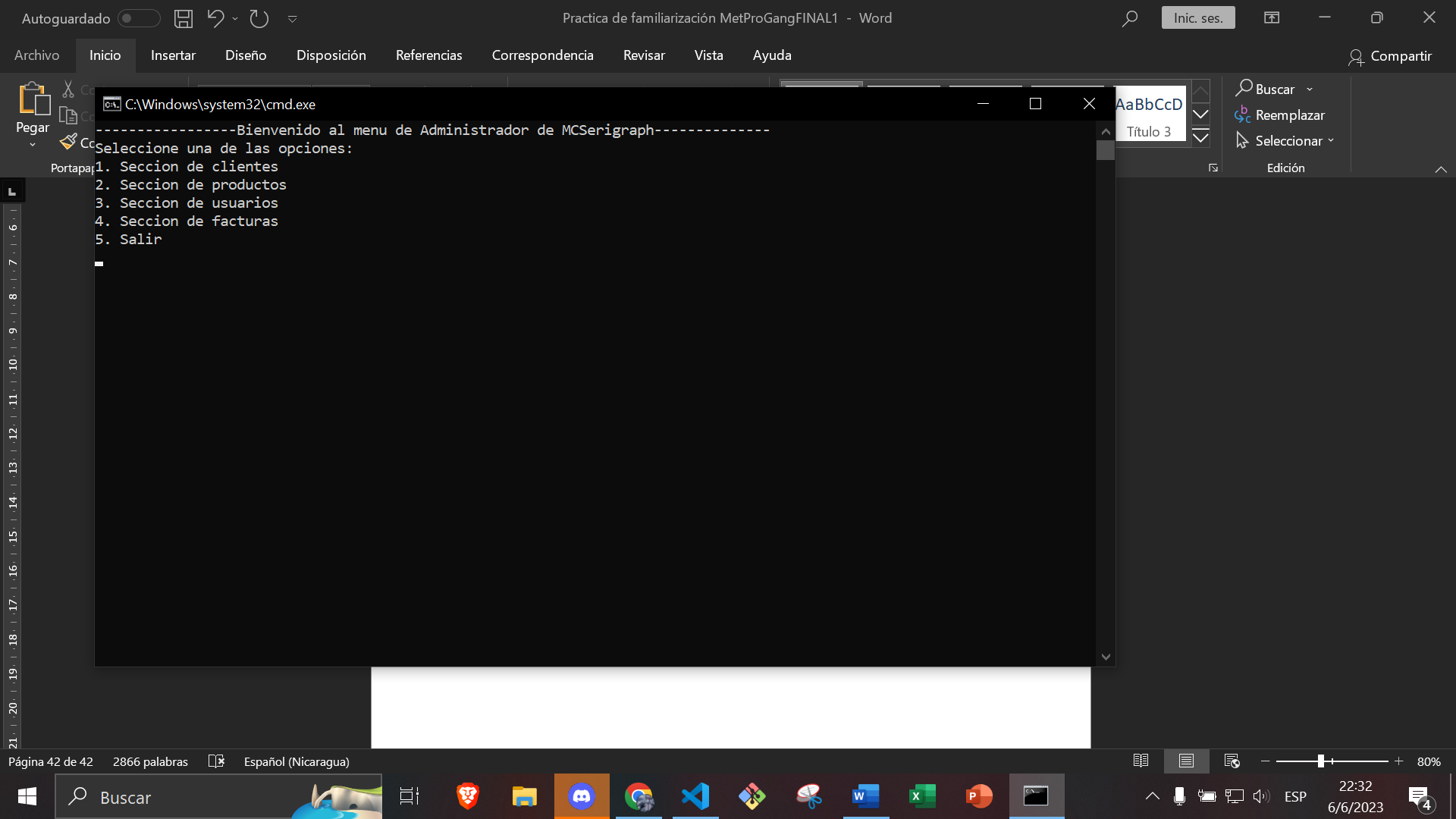
producto productos[MAX];

## Código fuente y salidas

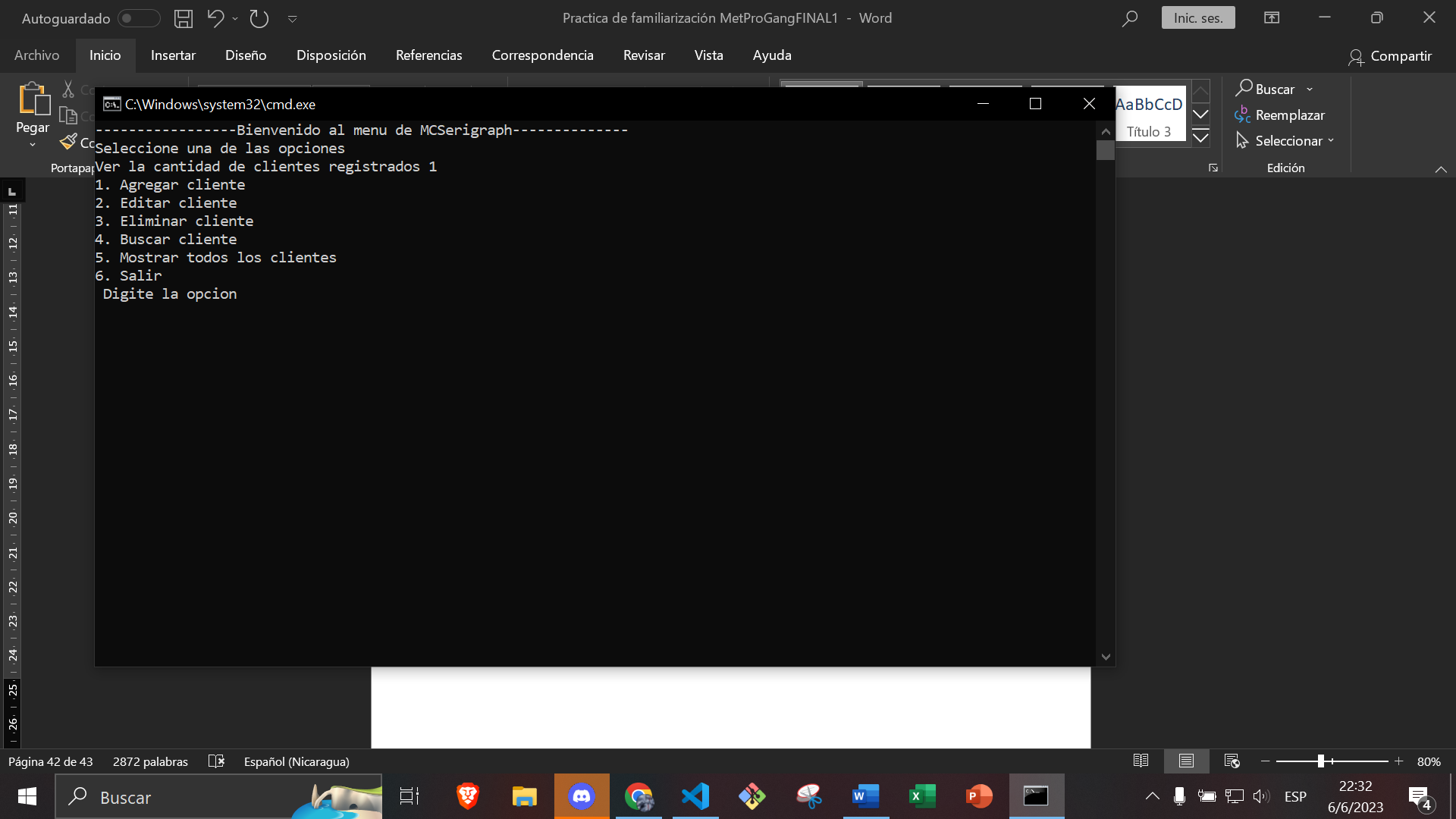
**MENU PRINCIPAL**



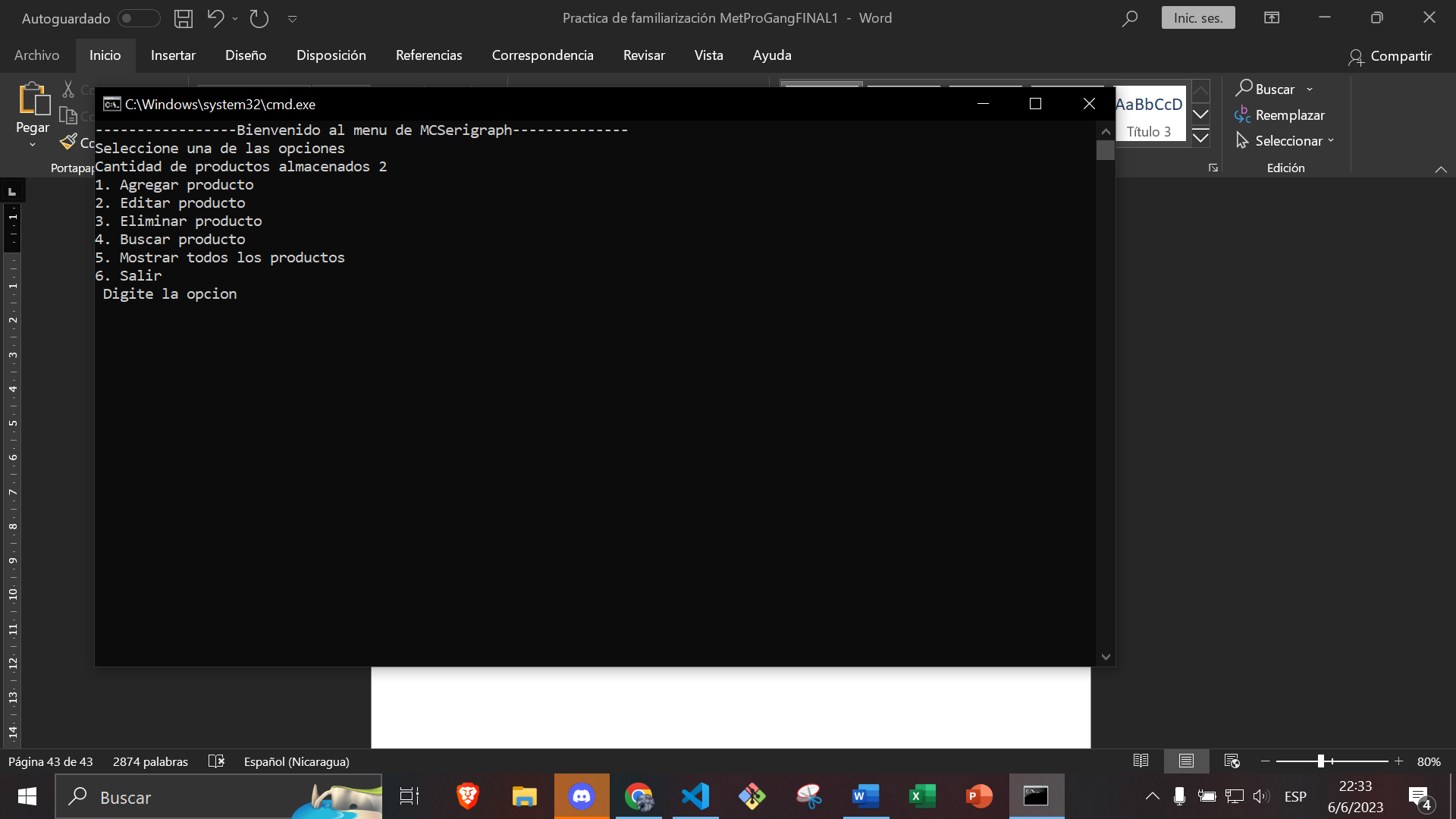
**MENU ADMIN CON TODAS LAS FUNCIONES**



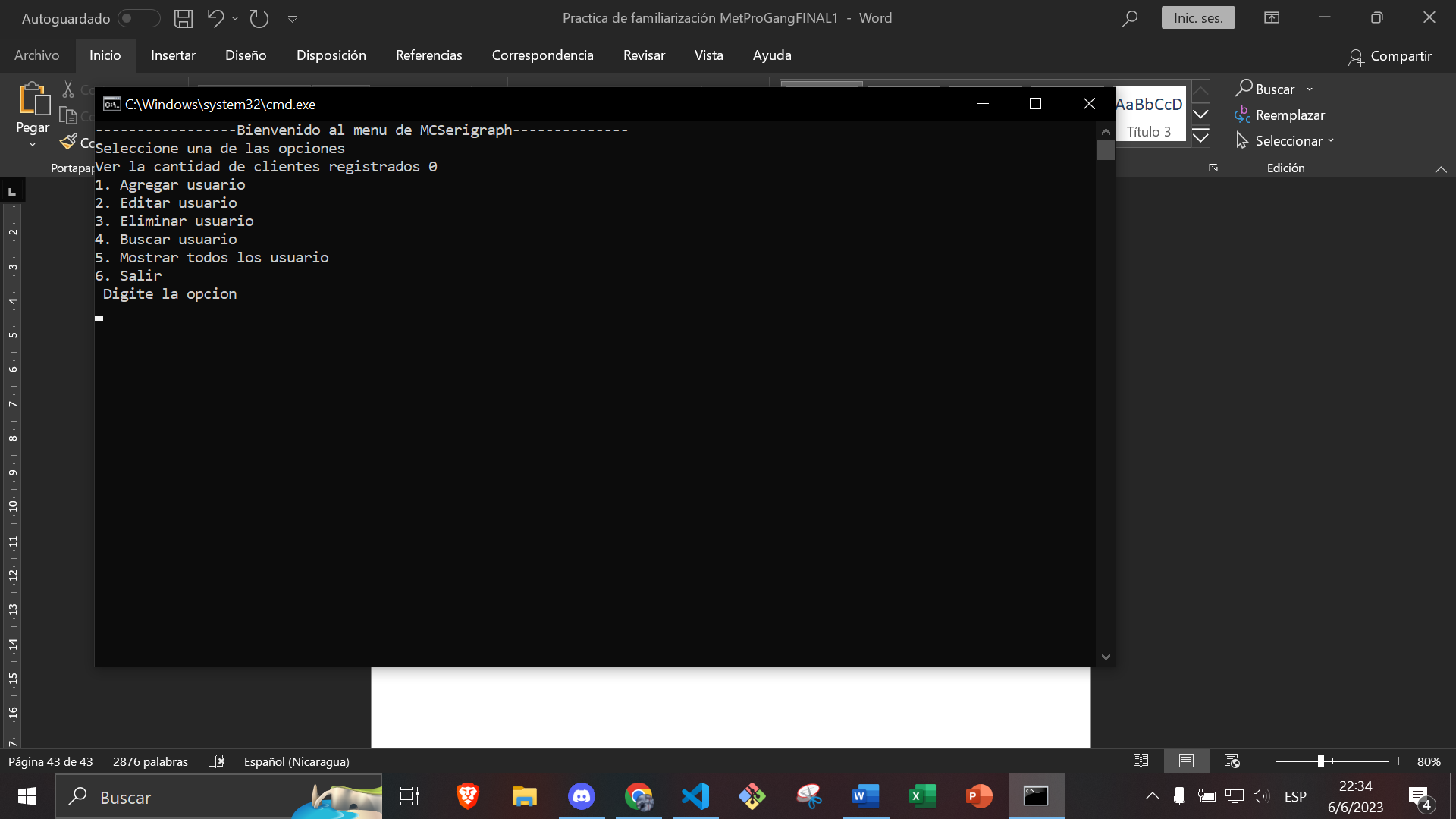
**MENU CLIENTES**



**MENU PRODUCTOS**



**MENU USUARIOS**



**MENU FACTURAS**

