ESIZON

METODOLOGIA DE LA PROGRAMACION

Torres Diez, Pablo

Practica Modularidad

Indice

Introducción

Este programa es la implementación de una versión simplificada de un portal de compras online, donde tenemos a diferentes perfiles:

El usuario común como cliente, que realizara compras, seguimiento de pedidos y devoluciones. Los vendedores donde tenemos al proveedor principal, los cuales son los propietarios de la aplicación, ESIZON, y los secundarios, que pueden ser diferentes proveedores.

- Las principales características del programa por parte de los proveedores (ESIZON y otros) es la gestión, venta y envíos de productos, con funciones segundarios como la creación de vales y descuentos, y por parte de la logística, el envío a dirección del cliente o lockers, y la gestión del transportista
- Las principales características del programa por parte de los clientes es la vista del catálogo y compra de productos, teniendo la opción de seguimientos de envíos a domicilio o lockers, y la devolución de los mismos, con la posibilidad de aplicar descuentos o vales y usando su cartera como método de pago.

En la documentación de este programa se distingue dos partes:

- Documentación interna: Documentación contenida en las líneas de comentarios del programa, que se podrá ver en el código.
- Documentación externa: Documentación que recoge los diferentes manuales para el usuario y su instalación, análisis, diagramas y pruebas del sistema de su correcto funcionamiento. Se diferencia dos partes: documentación del usuario y documentación del sistema.

Documentación de usuario

A continuación, se explicará todo lo relacionado con la funciones del sistema para el usuario, así como todos sus manuales.

Descripción funcional

Este programa esta escrito en lenguaje C, contine 13 módulos, en cada uno de ellos tiene su respectiva funciones y estructura que interactúan entre si para llevar a cabo un sistema de venta de productos.

La función principal, llamada menu(), es la que lleva a cabo el inicio del programa, carga y guarda la información y alberga el menu principal de programa, donde podrá iniciar sesión, clientes o proveedores, y el registro de nuevos clientes. En función de lo opción elegida y del usuario que inicie sesión le llevara a diferentes menu, respetivas llamadas a función.

La función MenuClientes() es el menu para los clientes donde podrán acceder a su perfil, datos y modificaciones, al catalogo de productos, realizar compras y devoluciones.

La función MenuProveedores() es el menu de los proveedores donde podrán visualizar y modificar toda la información de productos y pedidos, únicamente, suyos.

La función MenuAdmin() es el menu de los administradores, o mejor dicho, el principal proveedor/propietario de la aplicación ESIZON, los cuales tendrá acceso a toda la información de la aplicación, que podrá gestionar y tienen la posibilidad dar de alta nuevos proveedores y gestionarlos.

Tecnología

Para el desarrollo del programa se han usado dos diferentes entornos de desarrollo:

- Code::Blocks: Version 20.3
- Visual Studio Code: Version 1.80.0
- Markdown: Via Visual Studio Code, pasandolo a World para crear la portada
- Trello: Gestión de trabajo para el desarrollo del programa
- Github Desktop: Actualizaciones del desarrollo y repositorio

Manual de instalación

Los requisitos del programa son:

- Si se ejecuta por el ejecutable (.exe) como mínimo el S.O: Windows XP (Service Pack 2)
- Si es necesario copilar para otros sistemas operativos será necesario la instalación de GNU Compiler para el sistema operativo deseado.

Para la instalación del programa nos bastara por ejecutarlo por primera vez (ESIZON.exe) y el propio programa inicializara el sistema, generando los ficheros necesarios. Para otros sistemas operativos, una vez copilado y ejecutado procederá al mismo procedimiento.

La primera vez que se ejecute el programa nos pedirá la creación de un usuario root, la cual nos solicitara una nueva contraseña y creara este usuario administrador, mostrara por pantalla los datos necesarios para iniciar sesión con dicho usuario.
Se recomienda la creación de un usuario cliente y proveedor para el correcto funcionamiento del programa.

Acceso al sistema

Para acceder al sistema podremos elegir dos opciones que nos mostrara en el primer menu nada más ejecutar el sistema una vez inicializado, la primera es Iniciar sesión donde con los datos, email y contraseña, la cual nos llevara al menu correspondiente, cliente, proveedor o administrador; la segunda opción es Registrarse, esta opción será válida solo para el usuario cliente ya que solo podremos crear tal usuario, los demás solo podrá dar de alta un administrador, después de completar el formulario podremos iniciar sesión como en la primer a opción.

Manual de referencia

Una vez iniciada la sesión tendremos 3 posibles perfiles según nuestros datos: Cliente, Proveedor o Administrador (Operador). Tendremos 3 menús diferentes:

Los Clientes tendrán acceso en su menú a 6 diferentes opciones:

- En la opción de *Perfil* podremos gestionar nuestra información, tanto visualizarla como modificarla.
- En la opción de *Productos* tendremos acceso al catálogo de productos que se ofertan.
- En la opción de *Descuentos* mostrara el listado de descuentos y vales obtenidos.
- En la opción de *Pedidos* es la función principal del sistema ya que podremos realizar las compras en la aplicación y gestionaremos nuestros pedidos, como envíos pendientes. Podrá aplicar descuentos y/o cheques. Podrá realizar pedidos a domicilio o usar la función de locker del sistema.
- En la opción de *Devolución* podremos gestionar devoluciones de productos de pedidos ya realizados.
- Por último podremos cerrar sesión y volver al menu principal.

Los Proveedores tendrán acceso en su menú a 4 diferentes opciones:

- En la opción de *Perfil*, al igual que los clientes, podrán visualizar y modificar sus datos.
- En la opción de *Productos* accederá a la información de sus productos que estén dados de alta en la plataforma, donde podrá realizar altas, bajas, búsquedas, listados y modificaciones de productos.
- En la opción de *Pedidos* accederá únicamente a la información de los pedidos de productos que él mismo suministra, donde podrá gestionar el estado de dichos pedidos de productos, asignar transportistas, lockers, etc.
- Por último podremos cerrar sesión y volver al menu principal.

Los Administradores tendrán acceso en su menu a 10 diferentes opciones:

- En la opción de *Perfil* visualizara su información y podrá modificar su contraseña.
- En la opción de *Clientes* podrá acceder a los datos de los clientes y podrá gestionar altas, bajas, búsquedas, listados y modificación de datos.
- En la opción de *Proveedores* al igual que en la opción de *Clientes* podrá gestionar la información de los proveedores del sistema, además con la opción exclusiva de poder dar de alta a nuevos proveedores.

- En la opción de *Productos* accederá a la información de los productos dados de alta en la plataforma y mediante su menú correspondiente podrá realizar altas, bajas, búsquedas, listados y modificaciones de productos.
- En la opción de *Categorías* accederá a la información de las categorías dadas de alta en la plataforma y mediante el menú correspondiente podrá realizar altas, bajas, búsquedas, listados y modificaciones de categorías, también podrá generar listados de productos por categoría.
- En la opción de *Pedidos* accederá a la información de los pedidos dados de alta en la plataforma, mediante el menú correspondiente podrá realizar altas, bajas, búsquedas, listados y modificaciones de pedidos. En los listados habrá que contemplar la posibilidad de listarlos según su estado. Esto es necesario para poder localizar rápidamente los pedidos cuya fecha de entrega sea próxima y que, por tanto, deben ser procesados con mayor prioridad. Otras opciones permitidas será la asignación de transportistas a los productos pedidos en función de la dirección del cliente y ciudad de reparto. Así como llevar a cabo la asignación de lockers a los pedidos en base a la localidad de entrega y ubicación del locker.
- En la opción de *Transportistas* accederá a la información de todos los transportistas dados de alta en la plataforma y mediante el menú correspondiente podrá realizar altas, bajas, búsquedas, listados y modificaciones de transportistas.
- En la opción de *Descuentos* accederá a la información de todos los códigos promocionales y/o cheques regalo dados de alta en la plataforma. Mediante el menú correspondiente podrá realizar altas, bajas, búsquedas, listados y modificaciones de estos descuentos Además también podrá generar los listados de clientes que tienen asignado algún cheque regalo/código promocional, así como los listados de clientes que han hecho uso de algún cheque regalo/código promocional. Podrá asignar a un cliente determinado un cheque regalo. Podrá crear descuentos o cheques aplicables a todos los usuarios.
- En la opción de *Devoluciones* gestionará las devoluciones de pedidos realizados por los clientes, donde podrá listar, gestionar y dar de alta nuevas solicitudes.
- Por último podremos cerrar sesión y volver al menu principal.

Guía del operador

- El administrador ser el primer usuario que inicie la aplicación y cree el usuario root.
- Gestionará la información total del sistema.
- Es el responsable de gestionar las altas de nuevos proveedores, ya que es unico usuario con esta opción.
- Es el responsable de gestionar los transportes y lockers, realizara la gestión de envios, dara de alta a los transportistas y la gestión de Lockers.

Documentación del sistema

A continuación, se explicará el funcionamiento de los módulos, su análisis y diseño y prueba del software.

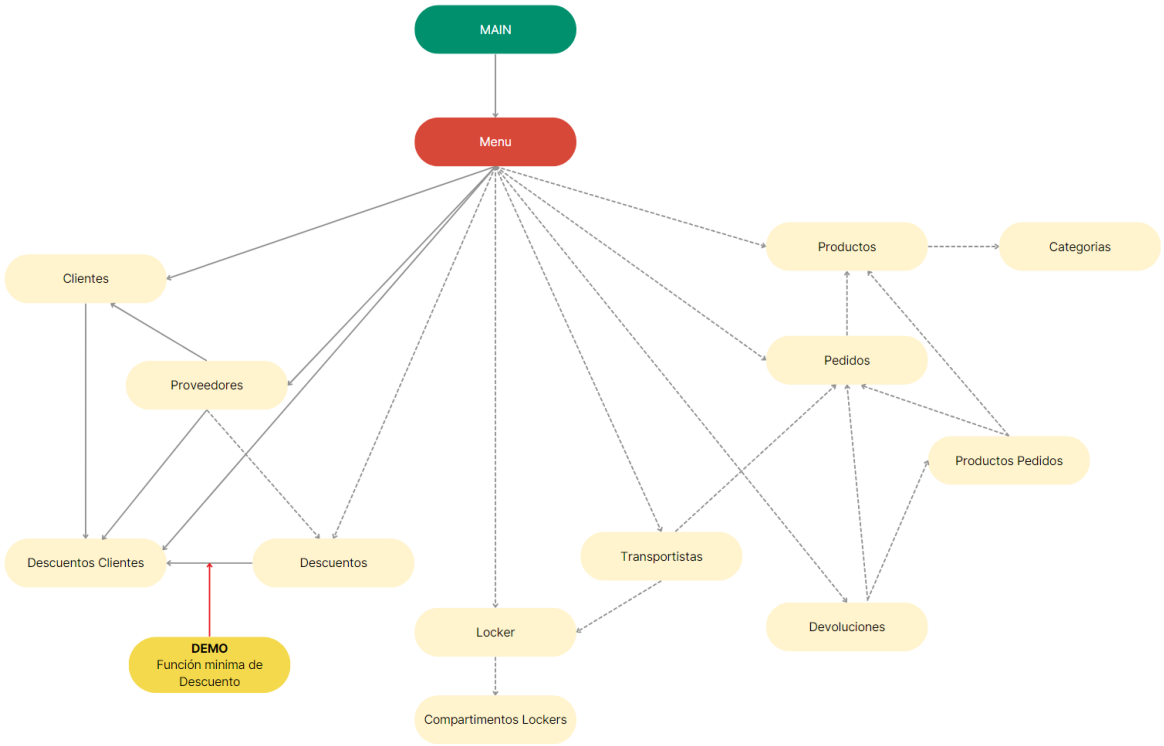
Especificación del sistema

Se nos plantea realizar la implementación de un sistema simplificado de una aplicación para comprar productos del proveedor ESIZON y otros proveedores adheridos. Tenemos tres tipos de usuarios, Cliente, el usuario normal, que podrá utilizar la aplicación para hacer uso de ella, Administrador, ESIZON, que será la encargada principal de la aplicación de su correcto funcionamiento de la gestión de productos de ESIZON, y administrara todos los datos de la aplicación. Los Proveedores, parecidos a los administradores, pero limitado, gestionara solo la información relacionada con sus productos.

Restricciones del programa:

- La aplicación también almacenará los datos de los clientes, proveedores, productos, categorías, descuentos, descuentos de clientes, lockers, compartimentos de lockers, pedidos, productos pedidos, transportistas y devoluciones.
- Solo puede haber de tres tipos de perfiles: cliente, administrador y proveedores.

A continuación, se representará gráficamente la estructura de los módulos del programa.



Módulos

- Módulo **Menu** contiene los menús de los diferentes usuarios y las llamadas a funciones necesario para que el programa funcione.
- Módulo **Clientes** gestiona la información de los clientes del sistema, donde se podrán dar de alta, baja, modificar, listar y buscar datos.
- Módulo **Proveedores** gestiona la información de los proveedores o administradores del sistema, donde se podrán dar de alta, baja, modificar, listar y buscar datos. Tambien alberga las funciones administrativas del sistema
- Módulo **Productos** *modulo sin desarrollo...*
- Módulo **Categorías** *modulo sin desarrollo...*
- Módulo **Descuentos** *modulo sin desarrollo...*
- Módulo **DescuentosClientes** Aplica los distintos descuentos y/o cheques del sistema a cada cliente compatible, y su estado y valides, gestiona toda información de los descuentos de cada cliente.
- Módulo **Lockers** *modulo sin desarrollo...*
- Módulo **CompartimentosLockers** *modulo sin desarrollo...*
- Módulo **Pedidos** *modulo sin desarrollo...*
- Módulo **ProductosPedidos** *modulo sin desarrollo...*
- Módulo **Transportistas** *modulo sin desarrollo...*
- Módulo **Devoluciones** *modulo sin desarrollo...*
- *Módulo **Demo** este modulo sustituye los módulos faltantes por si es necesario de su funcionamiento para los restantes*.

Plan de prueba

Las pruebas realizadas corresponderán solo a los módulos de Menu, Clientes, Proveedores y DescuentosClientes, excluyendo el *módulo demo* ya que no estaba previsto al inicio del desarrollo de la aplicación.

Prueba de Caja Blanca

```
//Cabecera: Función BuscarCliente()
//Precondicion: estructura cargada e inicializada, numero de clientes y recibir al menos de uno de estos datos: email, nombre o id
//Postcondicion: devuelve la posicion del cliente o -1 si no existe
int BuscarCliente(Cliente *arrayClientes, int n_clientes, char *nombre, char *email, int id, int op){

    int i, control = 0, posicion = -1, opcion;

    switch(op){
        case 1:
            for(i = 0; i <= n_clientes && control == 0; i++){
                if(strcmp(nombre, arrayClientes[i].Nomb_cliente) == 0){
                    printf("\nNombre: %s  Email: %s\nEs el cliente que busca? (1.-Si / 2.-NO):  ", arrayClientes[i].Nomb_cliente, arrayClientes[i].email);
                    scanf("%i", &opcion);
                    while(opcion != 1 && opcion != 2){
                        printf("\nError, valor no valido, vuelva a introducir (1.-Si / 2.-NO):  ");
                        scanf("%i", &opcion);
                    }
                    if(opcion == 1){
                        posicion = i;
                        control = 1;
                    }
                }
            }
            break;
        case 2:
            for(i = 0; i <= n_clientes && control == 0; i++){
                if(strcmp(email, arrayClientes[i].email) == 0){
                    posicion = i;
                    control = 1;
                }
            }
            break;
        case 3:
            for(i = 0; i <= n_clientes && control == 0; i++){
                if(id == arrayClientes[i].Id_cliente){
                    posicion = i;
                    control = 1;
                }
            }
            break;
    default: printf("\nError en la seleccion de modo de busqueda.");
    }

    //printf("\nError, no se ha encontrado ningun cliente");

    return posicion;
```

Figura 2: Código funioón BuscarCliente()

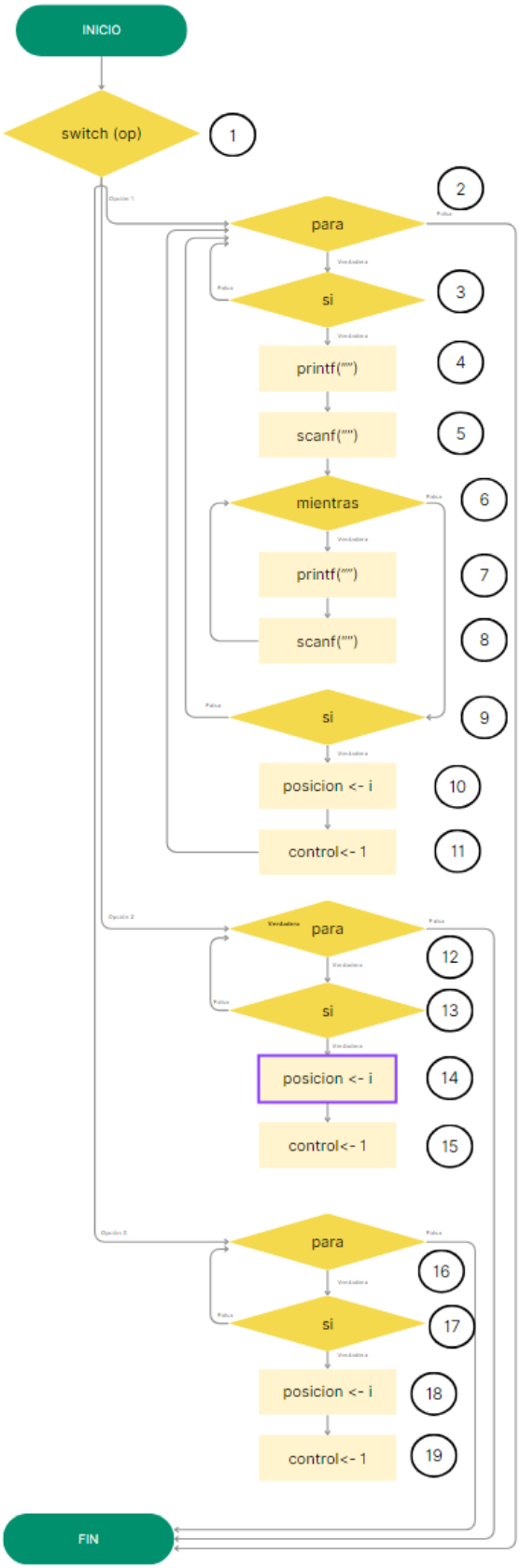


Figura 3: Diagrama de flujo función BuscarCliente()

Esta función tiene el uso de buscar a los clientes por 3 métodos, la primera por nombre, la segunda por email y la ultima por la id. La precondición de la función es que la estructura de Cliente este cargada e inicializada, debe contener datos, y según el op, método de búsqueda, elegido, ser reciba nombre, email o id.

En primer lugar, se desarrolla el diagrama de control de flujo para obtener las rutas independientes. Una vez tenemos las rutas independientes sacamos la complejidad ciclomática del CFD mediante las fórmulas:

- V(G) = NA – NN + 2 = 29 – 19 +2 = 12
- V(G) = NNP +1 = (8 + (1 x 3)) + 1 = 12
- V(G = numero de regiones = 12

Siendo NA (Número de aristas), NN (Número de vértices), NNP (Numero de nodos predicado)

Obtenemos que la complejidad ciclomática es 12 y las rutas independientes son:

Ruta 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 2

Ruta 2: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 2

Ruta 3: 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 11 – 2

Ruta 4: 1 – 2 – 3 – 4 – 5 – 6 – 9 – 2

Ruta 5: 1 – 2 – 3 – 2

Ruta 6: 1 – 2

Ruta 7: 1 – 12 – 13 – 14 – 15 – 12

Ruta 8: 1 – 12 – 13 – 12

Ruta 9: 1 – 12

Ruta 10: 1 – 16 – 17 – 18 – 19 – 16

Ruta 11: 1 – 16 – 17 – 16

Ruta 12: 1 – 16

Por tanto, según la postcondición que es que devuelva la posición del cliente buscado, por las rutas 1 , 7 y 10 se obtiene el resultado deseado, aunque si la función no logra su objetivo para cumplir la postcondición, en el caso de recibir una id, nombre o email que no exista en los datos o sea nulo devolvería la posición -1 que ayudaría al programa a identificar el problema, las rutas 2, 3, 4, 5, 6, 8, 9, 11 y 12.

Por lo tanto, nos aseguramos de que en todas las ocasiones que sea llamada la función si el resto del código está bien solo podrá ser llamada cuando la precondición sea cierta y podrá fallar por error humano (del administrador) y de igual manera funcionaria correctamente saliendo y dando la posibilidad de iniciar el proceso de búsqueda de los datos otra vez.

Prueba de Caja Negra

```
//Cabecera: Proveedor Funcion CargarProveedores(E/S entero: n_proveedores)
//Precondicion: Debe existir el fichero proveedores.txt
//Postcondicion: Carga la estructura de proveedores.txt y devuelve el numero de
preveedores y administradores registrados
Proveedor *CargarProveedores(int *n_proveedores){

    int n_lineas = 0, i;
    char perfil[13], *ad = "admin";
    Proveedor *proveedores;
    FILE *f;

    f = fopen("AdminProv.txt", "r");     //Abrimos archivo de proveedores.txt

    if(f == NULL) printf("\nError al abrir proveedores.txt");  //Comprobamos que se abre
y contenga los datos bien

    n_lineas = ContarLineas(f);     //numeros de lineas igual al nuemro de proveedores
    *n_proveedores = n_lineas;
    proveedores = (Proveedor*)calloc(n_lineas, sizeof(Proveedor));     //Memoria
dinamica estrucutura

    rewind(f);

    for(i = 0; i < n_lineas; i++){                      //Bucle que almacena los datos
de los proveedores uno a uno
        fscanf(f, "%d-", &proveedores[i].Id_empresa);
        fscanf(f, "%[^-]-", proveedores[i].Nombre);
        fscanf(f, "%[^-]-", proveedores[i].email);
        fscanf(f, "%[^-]-", proveedores[i].Contrasena);
        fscanf(f, "%[^\n]\n", perfil);
        fflush(stdin);
        if(strcmp(perfil, ad) == 0){
            proveedores[i].Perfil_usuario = 0;
        }
        else {
            proveedores[i].Perfil_usuario = 1;
        }
        proveedores[i].Eliminado = 0;
        fflush(stdin);
    }

    fclose(f);
    return proveedores;

}
```

Figura 4: Codigo función CargarProveedores

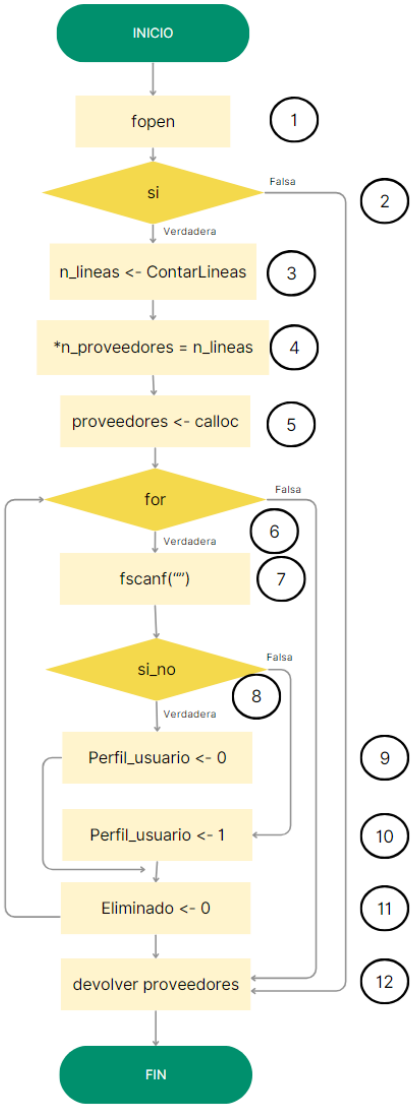


Figura 5: Diagrama de flujo de la función CargarProveedores

En la prueba de caja negra comprobaremos si tanto la precondición y postcondición se cumplen, mediante la entrada y salida de datos mediante los parámetros. El código en si no es tan importante que en la prueba de caja blanca ya que comprueba en todos los casos posibles con el diagrama de flujo de control, aquí nos importa la entrada y la salida y si es correcto el funcionamiento

Tenemos dos parámetros: **Proveedor, n_proveedores**

Proveedor es la declaración de es la estructura *Proveedor* la cual recibimos como encabezado de la función para obtener el formato de la estructura para posteriormente devolverlo a su *arrayproveedores* para guardar la información obtenida en el fichero *AdmProv.txt* y cargarla. En este caso, para el correcto funcionamiento la estructura debe estar declara en el respectivo *proveedores.h,* sabiendo que sin ese apartado no compilaría el código, este parámetro no afectara al funcionamiento de esta función si el programa ha copilado sin problemas y no se altera el formato de la estructura.

n_proveedores es el puntero al entero que contiene la cantidad de proveedores que van a ser almacenados en la estructura, es decir, la cantidad de proveedores en el fichero, lo cual lo capta por líneas escritas en él. Para el correcto funcionamiento de esta función no es necesario que contenga algún proveedor el dichero, es decir, puede tener valor 0, pero no puede ser inferior a 0, ya que la estructura *proveedores* no cargaría, aunque esto no va a suceder si no se fuerza ya que este dato se obtiene de las líneas del fiche que naturalmente será de líneas => 0, teniendo en cuenta que el valor de *n_proveedores* será mayor o igual que uno ya que el propio programa al inicializarse por primera vez tendrá un auto registro del primer proveedor/administrador. Este parámetro no afectara de forma negativa al funcionamiento de esta función ni del programa.

Por lo tanto, los parámetros recibidos serán correctos, y una vez pasados por la fusión, los parámetros de salida *n_proveedores* y *proveedores (arrayproveedores)* serán correctos. Devolviendo el número de proveedores en estructura y la estructura cargada.

Restos de pruebas

Las pruebas restantes no se ha podido realizar ya que por falta del resto de módulos de mis compañeros no he podido probar mas ni he tenido tiempo suficiente como para sustituir el trabajo faltantes de tales.

La prueba de módulos se ha realizado con Clientes – Proveedores, Clientes – DescuentoClientes, Cliente – DescuentoClientes – Proveedores, la cual es exitosa ya que entre los parámetros recibidos y devueltos entre si son correctos, el usuario realiza su correcto funcionamiento y muestra por pantalla los resultados esperados.

Documentación del código fuente

Toda la documentación interna sobre el detallado funcionamiento del programa estará escrita dentro del codigo.

Cabecera de menu:

Dentro del código del modulo menu.c escrito en la parte superior de cada función.

Cabecera de clientes:

Dentro del código del modulo clientes.c escrito en la parte superior de cada función.

Cabecera de proveedores:

Dentro del código del modulo proveedores.c escrito en la parte superior de cada función.

Cabecera de descuentos clientes:

Dentro del código del modulo descuentoclientes.c escrito en la parte superior de cada función.