

Proyecto sistema de ventas.

Objetivo:

Implementar un sistema que facilite al usuario la compra de productos y realizar diversas consultas, en una sucursal perteneciente a una cadena de almacenes. Igualmente facilitar la venta de productos por parte de las diversas sucursales y de la empresa.

Información general del desarrollo:

El programa fue realizado desde Dev C++ y Xcode, los cuales se utilizaron gracias a las comodidades que ambos App presentan al programar desde ellos.

Alcance:

El proyecto cuenta con las siguientes actividades:

- Análisis de los requerimientos del sistema a realizar.
- Estructuración del programa.
- Distribución de funciones básicas del programa a cada integrante del grupo.
- Construcción de estructuras para el sistema.
- Creación de archivos.
- Creación de las diversas funciones.
- Prueba de las funciones.
- Unión de funciones con el programa.

Tiempo de entrega:

Aproximadamente mes/ mes y medio desde la fecha de entrega del enunciado (18 de Octubre), a tal punto que el proyecto sea entregado el 18 de Noviembre del año en curso (2016).

Tabla de archivos

Archivo	Descripción
sucursales.txt	Contiene los nombres de las sucursales.
[1]Sr.pez.txt	Contiene los productos, presentaciones y cantidades de cada producto presente en esta sucursal.
[2]Donde Hamilton.txt	Contiene los productos, presentaciones y cantidades de cada producto presente en esta sucursal.

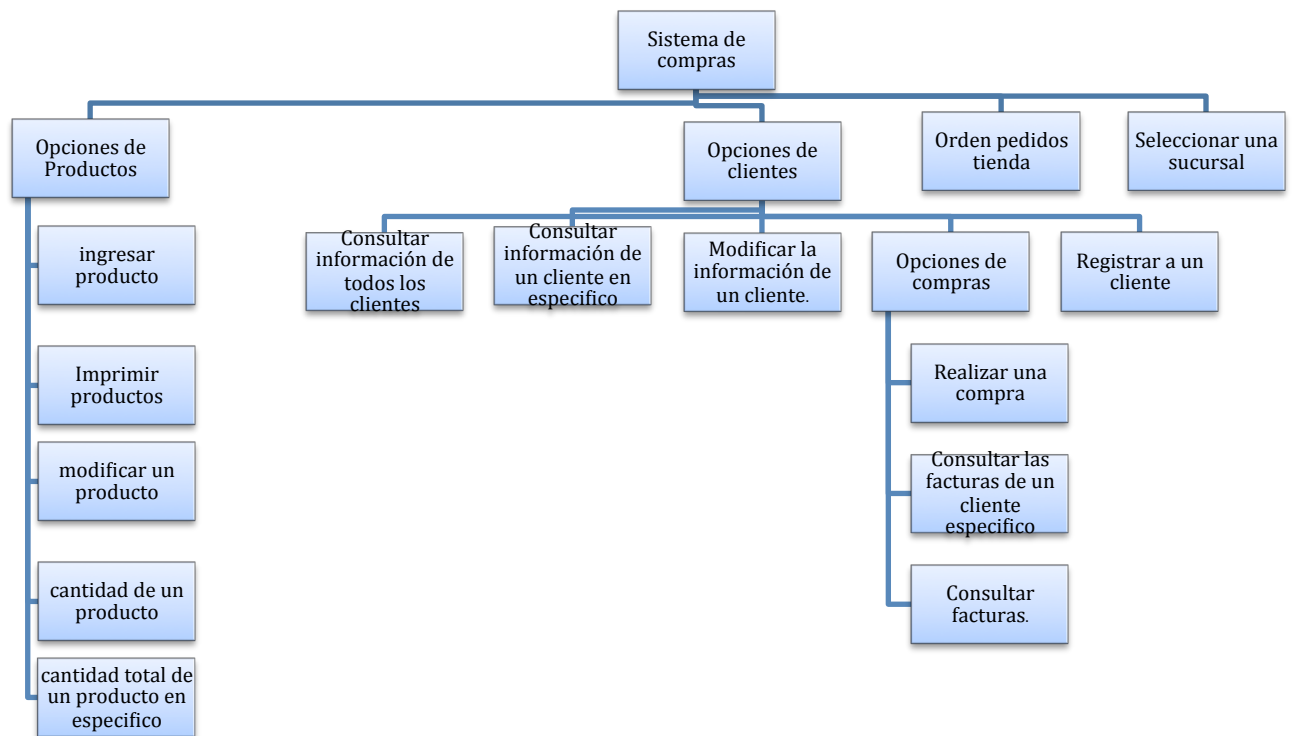
[3]Donde Lalis.txt	Contiene los productos, presentaciones y cantidades de cada producto presente en esta sucursal.
formato.txt	Contiene el formato de la factura para las ventas de las diversas sucursales.
tem.txt	Guarda de manera temporal los datos de los usuarios cuando se esta modificando la dirección de alguno de ellos.
clientes.txt	Contiene los datos de todos los usuarios (nombre, NIT, dirección).
temp.txt	Guarda de manera temporal los datos de los productos cuando se estan modificando los datos de alguno de estos.
codigos.txt	Contiene los nombres de los productos y su respectivo codigo de identificación.
facturas.txt	Contiene el nombre de los diferentes archivos en los cuales se encuentran las facturas de las compras realizadas por los diversos clientes.

Tabla de estructuras.

Estrcutura	Descripción
cliente	Contiene los datos esenciales de un cliente (Nombre, NIT y dirección).
presentación	Contiene los datos (tipo, existencias y costo) de una presentación especifica de un producto.
producto	Es una lista que contiene las diversas presentaciones que un producto posee.
sucursal	Es una lista donde se van guardando temporalmente los datos de los diversos productos registrados en una sucursal especifica.

compra	Contiene los datos (nombre del producto, presentación, cantidad, valor unitario, valor con IVA y el código de identificación del producto) de un ítem que se desea comprar.
listcompra	Es una lista donde se van guardando temporalmente los ítems que el cliente desea comprar.

Mapa de navegación



Definición de funciones:

Función	Descripción
producto presentacionproducto(producto nuevo);	Registra presentaciones de un producto en específico.
bool buscarproducto(char* nombre,char*tienda);	Busca un producto en una sucursal específica.
void imprimirproductos(char* sucursal);	Imprime todos los productos junto con sus presentaciones de una sucursal específica

char *seleccionarsucursal();	Retorna el nombre de la sucursal que selecciono el usuario.
void menu();	Muestra el menu de las opciones generales del programa.
void ingresarprod(char* sucursal);	Agrega un producto a una sucursal en especifico.
void modificarproductos(char* sucursal);	Modifica un producto de manera manual, en otras palabras, el usuario elige si quiere modificar o no el producto.
void venta(char *sucursal);	Pide al usuario los productos, presentaciones y cantidades que desea comprar.
void registrarcliente();	Añade un nuevo cliente al archivo donde estos se encuentran.
bool buscarcliente(char *nombre, char *archivo);	Busca un cliente en el archivo donde se encuentran los clientes, retorna verdadero si encuentra el cliente, de lo contrario retorna falso.
void infoclientes();	Imprime la información de todos los clientes que se encuentran registrados en el archivo "clientes.txt".
int cantidadproductototaltienda(char* sucursal,char* nombreprod);	Retorna la cantidad de un producto especifico en una tienda especifica.
int cantidadproductototal (char* nombreprod);	Retorna la cantidad de un producto especifico en todas las sucursales.
void cantidadproductosportienda(char* sucursal);	Retorna la cantidad de productos que tiene una tienda en especifico.
void ordenpedidostiendas();	Genera una orden .
void modificarinfo();	Permite la modificación de la información de un cliente en el archivo "clientes.txt".
int menuclien();	Retorna el valor de la acción que se quiere ejecutar del menú (consultar informacion de todos los clientes, consultar informacion de un cliente en especifico, modificar informacion (direccion) de un cliente, opciones de compras y registrar cliente).
void clientes(char *sucursal);	Contiene la relación del menuclien() con las funciones que ejecutan cada una de las acciones presentadas en el menú.

<code>void infoclienteespe();</code>	Imprime la información de un cliente en específico.
<code>void productos(char* sucursal);</code>	Contiene el menu con relación a las funciones de los productos (ingreso productos, imprimir productos, modificar productos, cantidad de un producto, cantidad total de un producto específico).
<code>int menuventa();</code>	Cuenta con las opciones que guardan relacion con la venta (Realizar compra, consultar Todas las facturas y consultar las facturas de un cliente específico) y retorna el valor de la opción que el usuario quiere realizar.
<code>void ven(char * sucursal);</code>	Permite la conexión entre el menuventa() y las funciones que cumplen con el funcionamiento enunciado en el menu.
<code>char *rev(char*frase);</code>	Voltea una frase y la retorna.
<code>void consultafacturas();</code>	Permite la consulta de todas las facturas que hay en el sistema.
<code>void consultarfacclien(char *clien);</code>	Permite la consulta de las facturas de un cliente en específico.
<code>listcompra *crear();</code>	Crea la lista donde se colocarán los productos que se quieren comprar.
<code>bool vacia(listcompra * lista);</code>	Revisa si la lista donde se van a guardar los productos esta vacia, en caso de que sea asi retorna verdadero.
<code>void insertar(listcompra * lista, struct compra infoNueva);</code>	Inserta un producto en la lista de productos por comprar.
<code>cliente obtinfo(char *nombre);</code>	Retorna la información de un cliente en específico.
<code>int preciouni(char *prod, char *pres, char*sucur);</code>	Retorna el precio unitario de un producto.
<code>float precioconIVA(int preuni, int cant);</code>	Calcula cual es el IVA de un producto específico que tiene que pagar el cliente. Y finalmente retorna este valor.
<code>void carrito(listcompra *lista, cliente datos, char *fecha);</code>	Genera la factura de la compra, teniendo como referencia la lista de productos comprados y los datos del cliente.
<code>void registrarproductocodigo(char *producto, long codigo);</code>	Agrega el codigo y nombre de un producto al archivo "codigos.txt".

Notas:

- Solo se permite la modificación de la dirección del cliente debido a que el NIT y nombre son constantes en la cotidianidad, único para cada cliente y además sirven de llave de acceso.

Integrantes del grupo:

- Daniel Hamilton-Smith Santa Cruz
- Laura Juliana Mora Páez