

TP FINAL LAB 2

2022

E-COMMERCE

ESTRUCTURA

LOGIN

USUARIO

CARRITO DE COMPRAS

VER PRODUCTOS

MIS DATOS DE ENVIO

SALDO DISPONIBLE

HISTORIAL COMPRA

CANCELAR COMPRA

Admin

VENTAS REALIZADAS

CANCELAR VENTA

MENU USUARIOS

MENU PRODUCTOS

MENU ENVIOS

ESTRUCTURA

LOGIN

USUARIO

Admin

VER PRODUCTOS

-VER LISTA DE PRODUCTOS

MENU USUARIOS

- VER USUARIOS
- VER INFO. USUARIO X ID
- CREAR USUARIO
- MODIFICAR DATOS ACCESO
- MODIFICAR ESTADO Y SALDO

SALDO DISPONIBLE

-VER SALDO DISPONIBLE

MENU PRODUCTOS

- STOCK DISPONIBLE
- BUSCAR PRODUCTO X NOMBRE
- AGREGAR PRODUCTO
- MODIFICAR PRODUCTO

CANCELAR COMPRA

-CANCELAR POR ID DE COMPRA

MENU ENVIOS

- ENVIOS PENDIENTES
- CONFIRMAR ENVIO
- CANCELAR ENVIO

E-COMMERCE

ESTRUCTURA



LOGIN

USUARIO

CARRITO DE COMPRAS

ESTRUCTURAS

```
typedef struct{
    producto dato;
    int cantidad;
    float subtotal;
    int flag;
} subVenta;
```

```
typedef struct{
    char nombre[30];
    char nombreCategoria[30];
    char descripcion[100];
    float precioVenta;
    float precioCosto;
    int nroCategoria;
    int cantidad;
    int flagStock;
} producto;
```

VER PRODUCTOS

ESTRUCTURAS

```
typedef struct{
    producto dato;
    struct nodoProductoD
    *siguiente;
    struct nodoProductoD
    *anterior;
} nodoProductoD;
```

```
typedef struct{
    char nombre[30];
    char nombreCategoria[30];
    char descripcion[100];
    float precioVenta;
    float precioCosto;
    int nroCategoria;
    int cantidad;
    int flagStock;
} producto;
```

Admin

VENTAS REALIZADAS

ESTRUCTURAS

```
typedef struct{
    venta dato;
    struct nodoVentaD siguiente;
    struct nodoVentaD anterior;
} nodoVentaD;
```

```
typedef struct{
} subVenta;
```

```
typedef struct{
} destino;
```

```
typedef struct{
    subVenta arreglo[50];
    destino despachar;
    float total;
    int idVenta;
    int idCliente;
    int estadoEnvio;
    int estadoVenta;
} venta;
```

CANCELAR VENTA

ESTRUCTURAS

```
typedef struct{
    venta dato;
    struct nodoVentaD siguiente;
    struct nodoVentaD anterior;
} nodoVentaD;
```

```
typedef struct{
    char pais[50];
    char provincia[50];
    char ciudad[50];
    char direccion[50];
} destino;
```

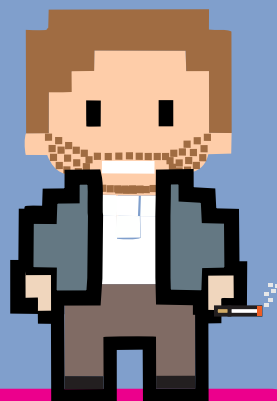
```
typedef struct{
    subVenta arreglo[50];
    destino despachar;
    float total;
    int idVenta;
    int idCliente;
    int estadoEnvio;
    int estadoVenta;
} venta;
```

```
typedef struct{
    producto dato;
    int cantidad;
    float subtotal;
    int flag;
} subVenta;
```

E-COMMERCE

ESTRUCTURA

LOGIN



USUARIO

MIS DATOS DE ENVIO

ESTRUCTURAS

```
typedef struct {
    char pais[50];
    char provincia[50];
    char ciudad[50];
    char direccion[50];
} destino;
```

SALDO DISPONIBLE

ESTRUCTURAS

```
typedef struct {
    subVenta arreglo[50];
    destino despachar;
    float total;
    int idVenta;
    int idCliente;
    int estadoEnvio;
    int estadoVenta;
} venta;
```

```
typedef struct {
    char nombre[30];
    char nombreCategoria[30];
    char descripcion[100];
    float precioVenta;
    float precioCosto;
    int nroCategoria;
    int cantidad;
    int flagStock;
} producto;
```

```
typedef struct {
    char username[30];
    char password[30];
    venta compras[30];
    subVenta carrito[50];
    int validosCarrito;
    int validosCompras;
    float saldo;
    int admin;
    int estadoCliente;
} usuario;
```

```
typedef struct {
    producto dato;
    int cantidad;
    float subtotal;
    int flag;
} subVenta;
```

Admin

MENU USUARIOS

ESTRUCTURAS

```
typedef struct {
    usuario dato;
    struct nodoProductoD *siguiente;
    struct nodoProductoD *anterior;
} nodoUsuarioD;
```

```
typedef struct {
    producto dato;
    struct nodoProductoD *siguiente;
    struct nodoProductoD *anterior;
} nodoProductoD;
```

```
typedef struct {
    char username[30];
    char password[30];
    venta compras[30];
    subVenta carrito[50];
    int validosCarrito;
    int validosCompras;
    float saldo;
    int admin;
    int estadoCliente;
    int idCliente;
} usuario;
```

```
typedef struct {
} producto;
```

MENU PRODUCTOS

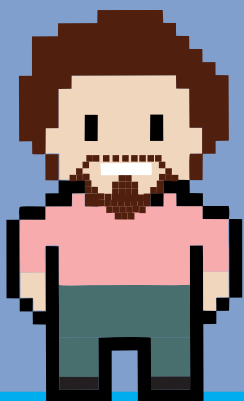
ESTRUCTURAS

```
typedef struct {
    producto dato;
    struct nodoProductoD *siguiente;
    struct nodoProductoD *anterior;
} nodoProductoD;
```

```
typedef struct {
    char nombre[30];
    char nombreCategoria[30];
    char descripcion[100];
    float precioVenta;
    float precioCosto;
    int nroCategoria;
    int cantidad;
    int flagStock;
} producto;
```

E-COMMERCE

ESTRUCTURA



LOGIN

USUARIO

HISTORIAL COMPRA

ESTRUCTURAS

```
typedef struct{
    venta dato;
    struct nodoVentaD siguiente;
    struct nodoVentaD anterior;
} nodoVentaD;
```

```
typedef struct{
    char pais[50];
    char provincia[50];
    char ciudad[50];
    char direccion[50];
} destino;
```

```
typedef struct{
    subVenta arreglo[50];
    destino despachar;
    float total;
    int idVenta;
    int idCliente;
    int estadoEnvio;
    int estadoVenta;
} venta;
```

```
typedef struct{
    producto dato;
    int cantidad;
    float subtotal;
    int flag;
} subVenta;
```

CANCELAR COMPRA

ESTRUCTURAS

```
typedef struct{
    venta dato;
    struct nodoVentaD siguiente;
    struct nodoVentaD anterior;
} nodoVentaD;
```

```
typedef struct{
    char pais[50];
    char provincia[50];
    char ciudad[50];
    char direccion[50];
} destino;
```

```
typedef struct{
    subVenta arreglo[50];
    destino despachar;
    float total;
    int idVenta;
    int idCliente;
    int estadoEnvio;
    int estadoVenta;
} venta;
```

```
typedef struct{
    producto dato;
    int cantidad;
    float subtotal;
    int flag;
} subVenta;
```

Admin

MENU ENVIOS

ESTRUCTURAS

```
typedef struct{
    char pais[50];
    char provincia[50];
    char ciudad[50];
    char direccion[50];
} destino;
```

E-COMMERCE

ROAD MAP

PROBLEMAS

GRAN RETRASO POR ERROR EN COMPILACION.

AL UTILIZAR GOTOXY PARA EL DISEÑO DEL MENU, CUANDO EL USUARIO AGRANDA LA VENTANA O SCROLLERBA LA VENTANA, EL MENU SE ROMPE.

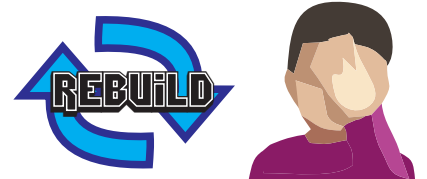
AL GENERAR EL MENU UNA DIMENSION FIJA NO SE PUEDE GENERAR TEXTO CON SCROLL HACIA ABAJO.

NO ENTENDEMOS PORQUE EXPLOTA CUANDO RECORREMOS LISTAS UTILIZANDO UNA FUNCION DE OBTENER USUARIO POR ID.

EN UN PRINCIPIO PENSABAMOS TRATAR EL CARRITO DE COMPRAS COMO UN ARCHIVO APARTE "CARRITOCOMPRAS.DAT", LUEGO NOS PERCATAMOS DE QUE ESTE CARRITO SERIA COMPARTIDO ENTRE LOS DISTINTOS USUARIOS, LO CUAL NO ERA NUESTRO OBJETIVO.

POR ALGUNA RAZON CUANDO ABRIMOS UN ARCHIVO DE FORMA RECURSIVA X CANTIDAD DE VECES EL PROGRAMA CRASHEA. (ABRIENDO Y CERRANDO EL ARCHIVO CORRECTAMENTE)

SOLUCIONES



TUVIMOS QUE PONER UN TAMANO DE VENTANA FIJO Y DESHABILITAR LA MAXIMIZACION DE LA VENTANA.

TUVIMOS QUE CREAR UN MENU QUE GENERARA "PAGINAS" AUTOMATICAMENTE Y ASI PODER MOSTRAR LA INFORMACION COMPLETA.

TRATAMOS DE UTILIZAR EL DEBUGGER PERO EXPLOTA TAMBIEN ASI QUE PROCEDEMOS A TRATAR DE DEBUQUEAR COMO ANTES. INVENTAMOS EL DEBUGGER CASERO:

```
SYSTEM("CLS");  
PRINTF("DEBUG");  
SYSTEM("PAUSE");
```

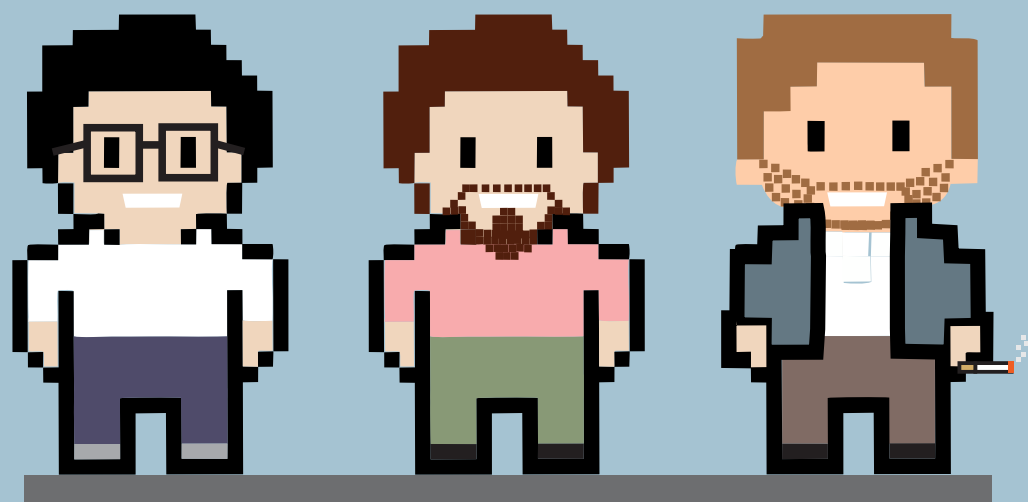
NOS DECIDIMOS POR INCLUIR EL CARRITO DE COMPRAS DENTRO DE LA ESTRUCTURA USUARIO PARA QUE ESTE SEA UNICO POR CADA USUARIO.

NOS DECIDIMOS POR MODIFICAR LAS FUNCIONES QUE CRASHEABAN PARA QUE ESTAS ABRIERAN EL ARCHIVO UNA UNICA VEZ.

AL CARGAR VARIOS USUARIOS,
APARENTEMENTE EL CODEBLOCKS NO
PUEDE MANEJAR EL ARCHIVO

AL CARGAR VARIOS USUARIOS,
APARENTEMENTE EL CODEBLOCKS NO
PUEDE MANEJAR EL ARCHIVO

TEAM PARTNERS

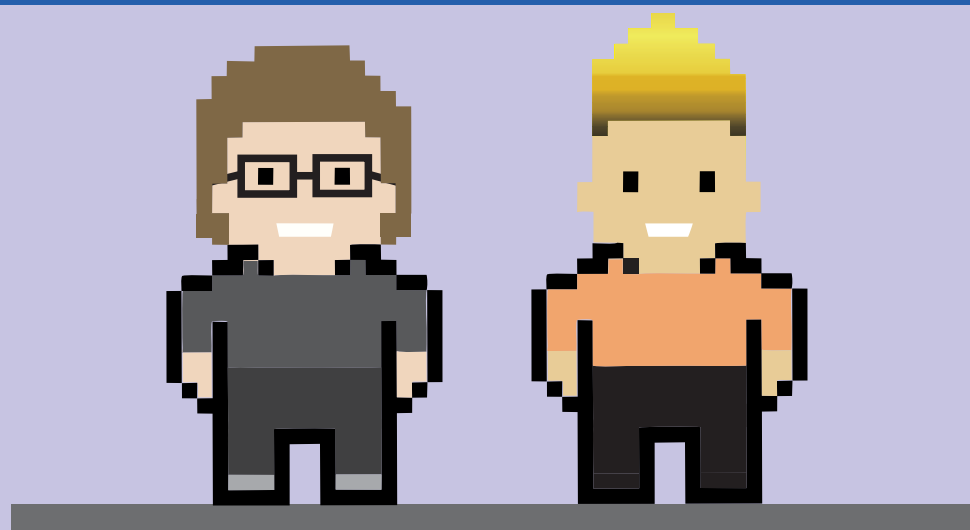


NAHUEL GARRIDO

NICOLAS LIEBY

JEREMIAS RICHARD

EQUIPO DOCENTE



GABRIEL CHALDU

MAURO GRIACA