**Proyecto Final de Estructura de Datos II**

**Profesor Ing Jose Carmona**

**Elaborado por:**

**David Martínez y Jorge Estrada**

# SuperSmart

Estructuras utilizadas y descripción de campos:

/\*Fin Estructuras de mulitilista\*/

struct ronda\_caja // estructura padre ronda doble para crear las cajas

{

int num\_caja; // número de la caja

char empleado; // empleado que atiende la caja

int num\_clientes; // cantidad de clientes haciendo fila para pagar

ronda\_caja \*der; // puntero derecha

ronda\_caja \*izq; // puntero izquierda

struct lista\_cliente \*abajo; //puntero que enlaza a estructura hijo de los clientes

};

struct lista\_cliente // estructura hijo bicola con acceso restringida para los clientes

{

int id\_cliente; // identificación del cliente

int estado; // estado del cliente

lista\_cliente \*ant; // puntero anterior

lista\_cliente \*sig; // puntero siguiente

struct lista\_carrito\_comprando \*abajo; // puntero que enlaza a estructura nieto del carrito de compra

};

/\*Fin Declaración de estructuras de mulitilista\*/

/\*Declaracion de estructura de datos auxiliares\*/

struct bicola\_carrito // bicola de carritos de compra libres

{

int num\_car; // número del carrito

bicola\_carrito \*cab1; // puntero inicio bicola por la izquierda

bicola\_carrito \*fin1; // puntero fin bicola por la derecha

bicola\_carrito \*cab2; // puntero inicio bicola por la derecha

bicola\_carrito \*fin2; // puntero fin bicola por la izquierda

bicola\_carrito \*der; // puntero para recorrer bicola por la derecha

bicola\_carrito \*izq; // puntero para recorrer bicola por la izquierda

bicola\_carrito \*extract\_b1(); // función de tipo bicola para extraer nodo por cab1

bicola\_carrito \*extract\_b2(); // función de tipo bicola para extraer nodo por cab2

};

struct bicola\_cliente\_espera // Bicola de clientes en espera de asignacion de carrito de compra

{

int id\_cliente; // identificación del cliente

int estado; // estado del cliente

bicola\_cliente\_espera \*cab1; // puntero inicio bicola por la izquierda

bicola\_cliente\_espera \*fin1; // puntero fin bicola por la derecha

bicola\_cliente\_espera \*cab2; // puntero inicio bicola por la derecha

bicola\_cliente\_espera \*fin2; // puntero fin bicola por la izquierda

bicola\_cliente\_espera \*der; // puntero para recorrer bicola por la derecha

bicola\_cliente\_espera \*izq; // puntero para recorrer bicola por la izquierda

bicola\_cliente\_espera \*extract\_b1\_ce(); // función de tipo bicola para extraer nodo por cab1

bicola\_cliente\_espera \*extract\_b2\_ce(); // función de tipo bicola para extraer nodo por cab2

};

struct ronda\_clientes\_comprando //estructura padre ronda doble de clientes comprando

{

int id\_cliente; // identificación del cliente

int estado; // estado del cliente

ronda\_clientes\_comprando \*der; // puntero de tipo ronda doble para recorrer por la derecha

ronda\_clientes\_comprando \*izq; // puntero de tipo ronda doble para recorrer por la izquierda

struct lista\_carrito\_comprando \*abajo; // puntro de tipo lista simple para enlazar con estructura hijo del carrito

};

struct lista\_carrito\_comprando // estructura hijo lista simple de carritos asignados a clientes

{

int num\_carrito; // número del carrito

lista\_carrito\_comprando \*sig; // puntero para recorrer lista simple

};

/\*Fin Declaracion de estructura de datos auxiliares\*/

Para visualizar correctamente el programa, favor configurar la consola de windows de la siguiente manera:

