Universidad Tecnológica Nacional Facultad Regional Avellaneda

**Laboratorio I**

Mateo Gaspar Geminiani

DNI 43815221

División 1B

## 1. Biblioteca “Núcleo”

* 1. **Funcion MainAltaCliente**

Realiza el alta de un Cliente llamando desde el main

eCliente arrayCliente.

Int tamCliente.

eLocalidad localidades.

Int tamLocalidad.

Int\* ultimoId.

Return. 0 en caso de no cargarse y 1 en caso de cargarse con éxito.

**int** **MainAltaCliente**(eCliente arrayCliente[], **int** tamCliente, eLocalidad localidades[], **int** tamLocalidad,**int**\* ultimoId);

**1.2 Funcion MainModificarCliente**

Realiza la modificación de dirección o localidad en un elemento del array de tipo eCliente llamando desde el main

eCliente arrayCliente.

Int tamCliente

eLocalidad localidades.

Int tamLocalidad

Return 1 si no modifico y 0 si modifico

**int** **MainModificarCliente**(eCliente arrayCliente[], **int** tamCliente, eLocalidad localidades[], **int** tamLocalidad);

* 1. **Funcion MainAltaPedido**

Realiza el alta de un elemento de tipo ePedido desde el main

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

Int\* ultimoId.

Return 0 si no puedo dar de alta y 1 si pudo darlo de alta.

**int** **MainAltaPedido**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos, **int**\* ultimoId);

# Funcion MainProcesarResiduo

Cambia el estado de algún elemento en “pendiente” del array pedidos a “completado” e ingresa las cantidades de plástico que fue procesado llamado desde el main.

ePedido pedidos.

Int tamPedidos.

Return 1

**int** **MainProcesarResiduo**(ePedido pedidos[], **int** tamPedidos);

# Funcion MainImprimirClientes

Imprime el listado de clientes y la cantidad de pedidos de recolección pendientes de cada uno.

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

**int** **MainImprimirClientes**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos);

# Funcion MainImprimirEstado

Imprime los pedidos del listado que tengan el mismo estado pasado por parámetro desde el main.

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

Int estado. -> Estado pasado por parámetro que va a imprimir

**int** **MainImprimirEstado**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos, **int** estado);

# Funcion MainPendientesEnLocalidad

Llamado desde el main imprime los pedidos que tengan el estado “pendiente” en una localidad.

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

eLocalidad localidades.

Int tamLocalidad.

Return 1

**int** **MainPendientesEnLocalidad**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos, eLocalidad localidades[], **int** tamLocalidad);

# Funcion MainPoliPromedio

Llamado desde el main imprime el promedio de polopropileno sobre el total de clientes.

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

Return 1

**int** **MainPoliPromedio**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos);

# Funcion MainClientesMasEstado

Llamado desde el main imprime el cliente mayor cantidad de pedidos en el estado pasado por parámetro.

eCliente arrayCliente.

Int tamCliente.

ePedido pedidos.

Int tamPedidos.

Int estadoPedido -> Pedido el cual va a mostrar que cliente tiene mas cantidad

Return 0 si no hay pedidos en el array pedidos que iguale con estadoPedido,

1 si existe algun pedido que iguale con estadoPedido

**int** **MainClienteMasEstado**(eCliente arrayCliente[], **int** tamCliente, ePedido pedidos[], **int** tamPedidos, **int** estadoPedido);

# Funcion CantidadEstado

Cuenta la cantidad de pedidos de un cliente que tengan el mismo estado que es pasado por parametro.

ePedidos pedidos

int tamPedidos

int idCliente -> id del cliente

int estadoPedido -> estado el cual va a devolver la cantidad

return Cantidad de pedidos en el iguales a estadoPedido

**int** **CantidadEstado**(ePedido pedidos[], **int** tamPedidos, **int** idCliente, **int** estadoPedido)

## 2. Biblioteca Cliente

**typedef** **struct**{

**int** idCliente;

**char** nombre[26];

**int** cuit;

**char** direccion[26];

**int** idLocalidad;

**int** isEmpty;

}eCliente;

**2.1** Funcion InicializarClientes

Inicializa el array de Clientes con la variable isEmpty en 1

eCliente arrayCliente.

Int tamCliente.

**void** **InicializarClientes**(eCliente arrayCliente[], **int** tamCliente);

**2.2** Funcion BuscarLibreCliente

Busca un elemento del array cliente que tenga la variable isEmpty en 1 y retorna su indice.

eCliente arrayCliente.

Int tamCliente.

Return -1 si no encontro espacio libre, distinto de -1 si encontro lugar libre

**int** **BuscarLibreCliente**(eCliente arrayCliente[], **int** tamCliente);

**2.3** Funcion CargaCliente

Carga un elemento del array cliente con los datos pasados por parámetro y la variable isEmpty a 0.

eCliente arrayCliente.

Int índice.

Char nombre.

Int cuit.

Char dirección.

Int localidad.

Int idCliente.

**void** **CargaCliente**(eCliente arrayCliente[], **int** indice, **char** nombre[], **int** cuit, **char** direccion[], **int** localidad, **int** idCliente);

**2.4** Funcion MainBajaCliente

Llamado desde el main realiza la baja de un cliente cambiando la variable isEmpty a 1.

eCliente arrayCliente.

Int tamCliente.

Return 0 si no pudo dar de baja y 1 si pudo dar de baja

**int** **MainBajaCliente**(eCliente arrayCliente[], **int** tamCliente);

**2.5** Funcion SolicitarIdCliente

Solicita el ingreso de un id de cliente existente y retorna su índice.

Char mensaje.

eCliente arrayCliente.

Int tamCliente.

**int** **SolicitarIdCliente**(**char**\* mensaje, eCliente arrayCliente[], **int** tamCliente);