

Universidad Tecnológica Nacional  
Facultad Regional Avellaneda

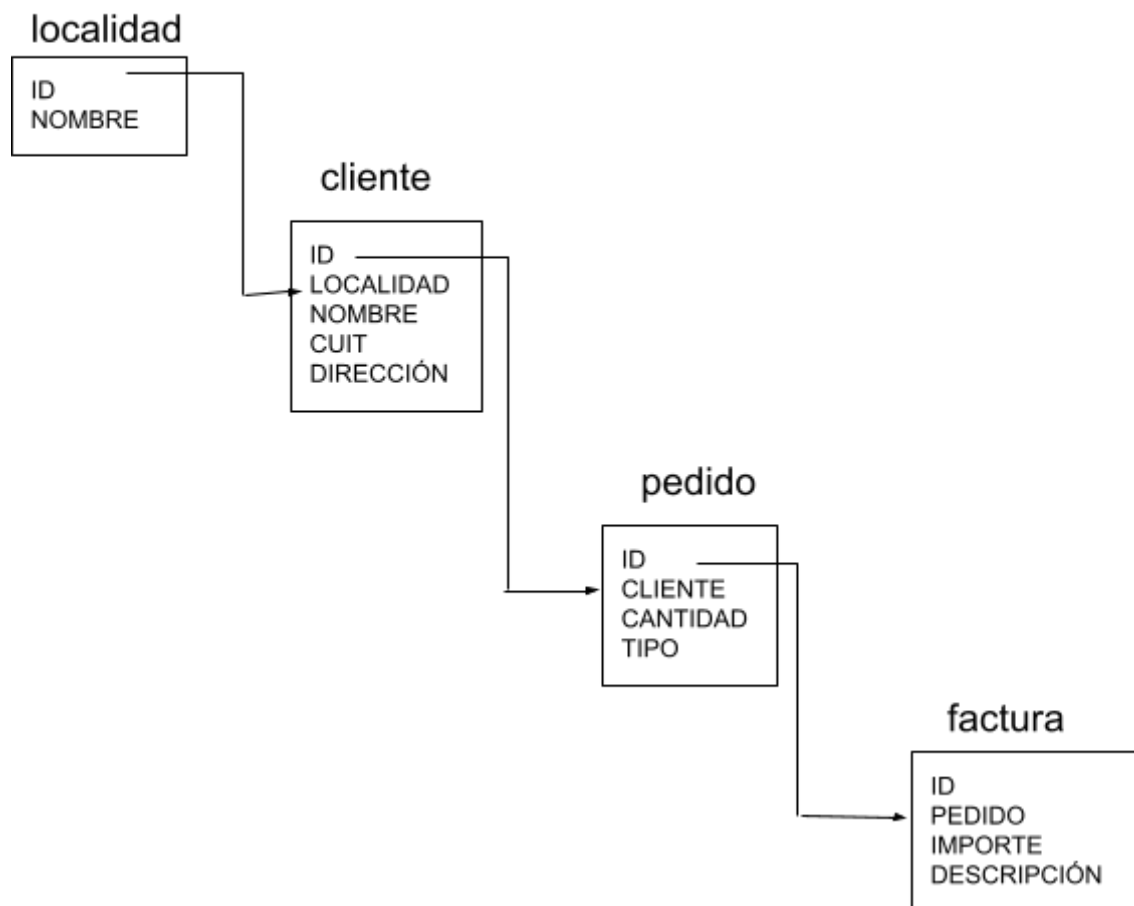
Recuperatorio Laboratorio  
Primer Parcial

Materia:  
Laboratorio de Programación I  
Estudiante:  
Marchan Gonzalez, Dayher Miguel.

## Introducción

El enunciado del presente informe se encuentra anexado al final de este documento. Se ha agregado una nueva entidad denominada “factura” la cual contiene información relacionada con la entidad “pedido”. Una vez que un pedido cambia su estado a COMPLETADO es creada una factura, conteniendo el ID del pedido, el importe según el campo cantidad del pedido, su propio ID, y una descripción sobre la factura. El uso de la nueva entidad se implementa en la opción del menú que permite encontrar la localidad con mayor importe acumulado en facturas, así como en la opción de procesar pedidos.

## Diagrama Entidad-Relación



Prototipos documentados:

```
/**
 * \brief Solicita datos al usuario para dar de alta un nuevo cliente
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \param lista de localidades
 * \param longitud de la lista de localidades
 * \return 0 en caso de éxito -1 en caso de error
 *
 */
```

```
int ingresarCliente(eCliente list[], int len, int id, eLocalidad list2[], int len2);
```

```
/**
 * \brief Se ingresa el ID del cliente. Luego se preguntará si se quiere confirmar la
eliminación.
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \return 0 en caso de éxito -1 en caso de error
 *
 */
```

```
int eliminarCliente(eCliente list[], int len);
```

```
/**
 * \brief Se ingresa el ID de cliente y se permitirá cambiar la dirección y la localidad.
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \return 0 en caso de éxito -1 en caso de error
 *
 */
```

```
int modificarCliente(eCliente list[], int len, eLocalidad list2[], int len2);
```

```
/**
 * \brief Crear pedido de recolección: Se imprimirán los clientes por pantalla y se pedirá que
se ingrese el ID de un cliente existente y la cantidad de kilos totales que se recolectarán del
cliente.
 * Se generará un ID para el pedido y el mismo quedará en estado "Pendiente" hasta que se
obtengan los residuos del cliente y se trasladen a la empresa.
 * \param lista de pedidos
 * \param longitud de la lista de pedidos
 * \param lista de Clientes
 * \param longitud de la lista de clientes
 * \return 0 en caso de éxito -1 en caso de error
 *
 */
```

```
int crearPedido(ePedido listaPedidos[], int nPedidos, int id, eCliente listaClientes[], int
nClientes);
```

/\*\*

\* \brief Procesar residuos: Se elegirá esta opción cuando los residuos que volvieron a la empresa se hayan procesado y separado en los diferentes tipos de plástico que la empresa puede reciclar.

\* Se imprimirán los pedidos por pantalla y se pedirá seleccionar el ID de uno de ellos.

\* Luego deberán ingresarse la cantidad de kilos de plástico de cada uno de los 3 tipos que la empresa puede procesar que se obtuvieron de los kilos totales que se recolectaron.

\* Por último, se marcará al pedido como "Completado".

\* \param lista de pedidos

\* \param longitud de la lista de pedidos

\* \param id para la nueva factura

\* \param lista de facturas

\* \param longitud de la lista de facturas

\* \return 0 en caso de éxito -1 en caso de error

\*

\*/

int procesarPedido(ePedido list[], int len, int idFactura, eFactura listaFacturas[], int nFacturas);

/\*\*

\* \brief Se imprimirá una lista de clientes con todos sus datos junto con la cantidad de "pedidos de recolección" que posee en estado "Pendiente".

\* \param lista de pedidos

\* \param longitud de la lista de pedidos

\* \param lista de clientes

\* \param longitud de la lista de clientes

\* \param lista de localidades

\* \param longitud de la lista de localidades

\* \return 0 en caso de éxito -1 en caso de error

\*

\*/

int imprimirClientes(ePedido listaPedidos[], int nPedidos, eCliente listaClientes[], int nClientes, eLocalidad listaLocalidades[], int nLocalidades);

/\*\*

\* \brief Se imprimirá una lista de los pedidos que se encuentren en estado "Pendiente" con la información: Cuit del cliente, dirección del cliente, cantidad de kilos a recolectar.

\* \param lista de pedidos

\* \param longitud de la lista de pedidos

\* \param lista de clientes

\* \param longitud de la lista de clientes

\* \return 0 de éxito -1 en caso de error

\*

\*/

int imprimirPendientes(ePedido listaPedidos[], int nPedidos, eCliente listaClientes[], int nClientes);

```

/**
 * \brief Ingresar una localidad e indicar la cantidad de pedidos pendientes para dicha
localidad.
 * \param lista de pedidos
 * \param longitud de la lista de pedidos
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \param lista de localidades
 * \param longitud de la lista de localidades
 * \return 0 en caso de exito -1 en caso de error
 *
 */

```

```

int imprimirPendientesPorLocalidad(ePedido listaPedidos[], int nPedidos, eCliente
listaClientes[], int nClientes, eLocalidad listaLocalidades[], int nLocalidades);

```

```

/**
 * \brief Imprime la cantidad de kilos de polipropileno reciclado promedio por cliente. (kilos
totales / cantidad de clientes)
 * \param lista de pedidos
 * \param longitud de la lista de pedidos
 * \param longitud de la lista de clientes
 * \return 0 en caso de exito -1 en caso de error
 *
 */

```

```

int promediarPropileno(ePedido listaPedidos[], int nPedidos, int nClientes);

```

```

/**
 * \brief Imprime el cliente con mas pedidos pendientes y completados.
 * \param lista de pedidos
 * \param longitud de la lista de pedidos
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \return 0 en caso de exito -1 en caso de error
 *
 */

```

```

int imprimirClienteConMasPedidos(ePedido listaPedidos[], int nPedidos, eCliente
listaClientes[], int nClientes);

```

```

/**
 * \brief Imprime la localidad con el mayor importe acumulado en facturas.
 * \param lista de pedidos
 * \param longitud de la lista de pedidos
 * \param lista de clientes
 * \param longitud de la lista de clientes
 * \param lista de localidades
 * \param longitud de la lista de localidades
 * \param lista de facturas
 * \param longitud de la lista de facturas

```

\* \return 0 en caso de éxito -1 en caso de error

\*

\*/

int imprimirLocalidadConMayorImporte(ePedido listaPedidos[], int nPedidos, eCliente listaClientes[], int nClientes, eLocalidad listaLocalidades[], int nLocalidades, eFactura listaFacturas[], int nFacturas);

### Link al video:

[Video-Defensa](#)

## Anexo

Una empresa de recolección y reciclado de plásticos requiere un sistema que les permita administrar sus clientes y pedidos de recolección. El programa contará con el siguiente menú:

- 1) **Alta de cliente:** Se da de alta un cliente con nombre de la empresa, cuit dirección y localidad. Se generará un ID único para este cliente que se imprimirá por pantalla si el alta es correcta.
- 2) **Modificar datos de cliente:** Se ingresa el ID de cliente y se permitirá cambiar la dirección y la localidad.
- 3) **Baja de cliente:** Se ingresa el ID del cliente. Luego se preguntará si se quiere confirmar la eliminación.
- 4) **Crear pedido de recolección:** Se imprimirán los clientes por pantalla y se pedirá que se ingrese el ID de un cliente existente y la cantidad de kilos totales que se recolectarán del cliente. Se generará un ID para el pedido y el mismo quedará en estado "Pendiente" hasta que se obtengan los residuos del cliente y se trasladen a la empresa.
- 5) **Procesar residuos:** Se elegirá esta opción cuando los residuos que volvieron a la empresa se hayan procesado y separado en los diferentes tipos de plástico que la empresa puede reciclar. Se imprimirán los pedidos por pantalla y se pedirá seleccionar el ID de uno de ellos. Luego deberán ingresarse la cantidad de kilos de plástico de cada uno de los 3 tipos que la empresa puede procesar que se obtuvieron de los kilos totales que se recolectaron. Por último, se marcará al pedido como "Completado".
- 6) **Imprimir Clientes:** Se imprimirá una lista de clientes con todos sus datos junto con la cantidad de "pedidos de recolección" que posee en estado "Pendiente".
- 7) **Imprimir Pedidos pendientes:** Se imprimirá una lista de los pedidos que se encuentren en estado "Pendiente" con la información: Cuit del cliente, dirección del cliente, cantidad de kilos a recolectar.
- 8) **Imprimir Pedidos procesados:** Se imprimirá una lista de los pedidos que se encuentren en estado "Completado" con la información: Cuit del cliente, dirección del cliente, cantidad de kilos reciclados de cada tipo de plástico.
- 9) Ingresar una localidad e indicar la cantidad de pedidos pendientes para dicha localidad.
- 10) Cantidad de kilos de polipropileno reciclado promedio por cliente. (kilos totales / cantidad de clientes)

Los tipos de plástico que la empresa puede reciclar son:

I



- **HDPE:** Polietileno de alta densidad (Envases para lácteos, perfumes, detergentes líquidos, etc.) •
- LDPE:** Polietileno de baja densidad (Bolsas de congelación de alimentos, tapas flexibles o bolsas de basura.)
- **PP:** Polipropileno (Plásticos utilizados en la industria automovilística y en la construcción.)

El resto de la basura recolectada es desechada

Continuar el examen previo:

- 1) Agregar la entidad Localidad.
- 2) Refactorizar las funciones que considere necesarias para que utilicen la nueva entidad.
- 3) Agregar los siguientes informes:
  - a) Cliente con más pedidos pendientes.
  - b) Cliente con más pedidos completados.
  - c) Cliente con más pedidos.