

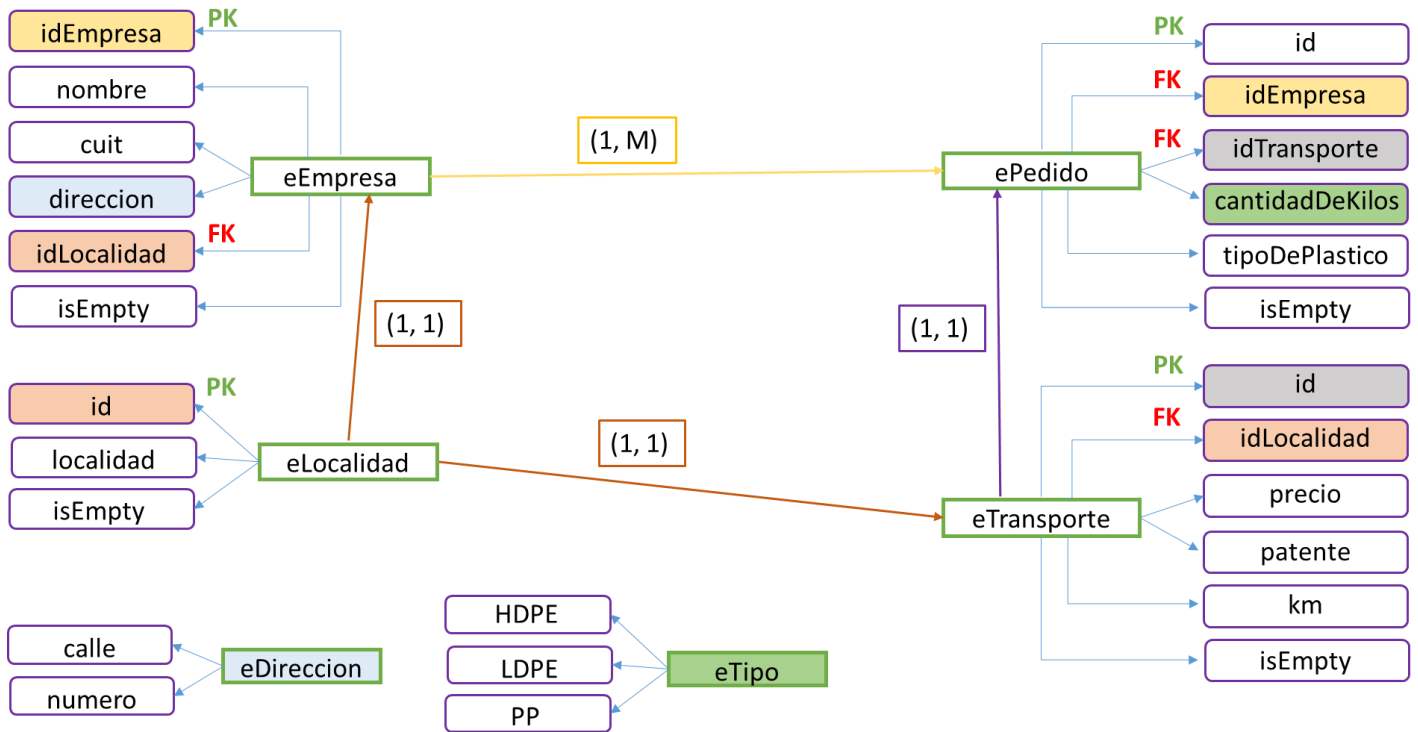


# RECUPERATORIO PRIMER PARCIAL LABORATORIO 1

## Datos de la alumna y la catedra

Galjot Maitena Camila  
DNI: 44457866  
Mail: galjotmaitena@gmail.com  
Division B

Materia: Laboratorio 1  
Profesor: Scarafilo German



### Enunciado del informe anexoado

13. Imprimir empresas con pedidos enviados, junto con la cantidad de pedidos enviados que posee, el total de km a recorrer para la realizacion de cada uno de los envios y el total a pagar por todos los envios.

### Prototipos de las funciones del nucleo del programa

#### Alta

```
/// \fn int Alta(eEmpresa[], int, eLocalidad[], int, int*)
```

```
/// \brief Da de alta una empresa
```

```
/// \param lista Array en la que se guardaran los datos de dicha empresa
```

```
/// \param tamaño tamaño del array
```

```
/// \param listaLocalidades Array de las localidades a elegir
```

```
/// \param tamañoLocalidad tamaño del array de localidades
```

```
/// \param idContador puntero que se utilizara para obtener el id de la nueva empresa
```

```
/// \return retorna (-1) si la lista es nula o no se encontro espacio disponible, y 0 si se pudo dar de alta una empresa
```

```
int Alta(eEmpresa lista[], int tamaño, eLocalidad listaLocalidades[], int tamañoLocalidad, int * idContador);
```

## Modificacion

```
/// \fn int Modificar(eEmpresa[], int, eLocalidad[], int)
/// \brief Modifica la direccion, localidad o ambas, de una empresa seleccionada por el usuario
/// \param listaEmpresas array de empresas de la que se obtendra la posicion de la empresa
requerida
/// \param tamañoEmpresas tamaño del array de empresas
/// \param listaLocalidades array de localidades, que serán mostradas para que el usuario
seleccione la nueva localidad
/// \param tamañoLocalidades tamaño del array de localidades
/// \return retorna 1 si la lista es nula, o 0 si se pudo realizar la modificacion

int Modificar(eEmpresa listaEmpresas[], int tamañoEmpresas, eLocalidad listaLocalidades[], int
tamañoLocalidades);
```

## Baja

```
/// \fn int Baja(eEmpresa[], int, eLocalidad[], int)
/// \brief Da de baja una empresa
/// \param listaEmpresas array de empresas en el que buscaremos a la empresa a la que se desea dar
de baja
/// \param tamañoEmpresas tamaño del array de empresas
/// \param listaLocalidades array de localidades necesario para mostrar las empresas que no se
dieron de baja
/// \param tamañoLocalidades tamaño del array de localidades
/// \return retorna 1 si las listas son nulas y 0 si se pudo dar de baja

int Baja(eEmpresa listaEmpresas[], int tamañoEmpresas, eLocalidad listaLocalidades[], int
tamañoLocalidades);
```

## Crear Pedido

```
/// \fn int CrearPedido(eEmpresa[], int, ePedido[], int, eLocalidad[], int, eTransporte[], int, int*, int*)
/// \brief Crea un pedido para una empresa
/// \param listaEmpresas array que se mostrara y del cual el usuario eligira una empresa que quiere
realizar el pedido
/// \param tamañoEmpresas tamaño del array de empresas
/// \param listaPedido array de pedidos, en el cual se buscare un espacio disponible y se guardara
el pedido
/// \param tamañoPedido tamaño del array de pedidos a recorrer
/// \param listaLocalidades array de localidades necesario para dar de alta el envio
/// \param tamañoLocalidades tamaño del array de localidades
```

```

/// \param listaTransportes array de transportes en el cual se buscara espacio para crear el envio
/// \param tamanoTransportes tamano del array de transportes
/// \param idTransporte puntero con el que obtendremos el id del envio
/// \param idPedido puntero con el que obtendremos el id del pedido
/// \return retorna -1 si las listas son nulas y 0 si se pudo crear el pedido correctamente

```

```

int CrearPedido(eEmpresa listaEmpresas[], int tamanoEmpresas, ePedido listaPedido[], int
tamanoPedido, eLocalidad listaLocalidades[], int tamanoLocalidades, eTransporte
listaTransportes[], int tamanoTransportes, int* idTransporte, int* idPedido);

```

## Procesar Pedido

```

/// \fn int ProcesarResiduos(ePedido[], int, eEmpresa[], int, eTransporte[], int)
/// \brief Completa el proceso del pedido, pidiendo el ingreso de los kilos de cada tipo de plastico
que vienen en el pedido
/// \param listaPedido array de pedidos en el que se buscara el pedido que quiero completar
/// \param tamanoPedido tamano del array de pedidos a recorrer
/// \param lista array de empresas necesario para mostrar la lista de pedidos
/// \param tamano tamano del array de empresas
/// \param listaTransportes array de transportes necesario para cambiar el estado del envio
/// \param tamanoTransportes tamano del array de transportes
/// \return retorna -1 si las listas son nulas o 0 si se pudo procesar correctamente el pedido
int ProcesarResiduos(ePedido listaPedido[], int tamanoPedido, eEmpresa lista[], int tamano,
eTransporte listaTransportes[], int tamanoTransportes);

```

## Imprimir clientes y la cantidad de pedidos pendientes

```

/// \fn int ImprimirClientesConPedidosPendientes(ePedido[], int, eEmpresa[], int, eLocalidad[], int)
/// \brief Muestra la lista de los clientes y la cantidad de pedidos que estan en estado pendiente
/// \param listaPedido array de pedidos que se recorrera en busca de los pedidos pendientes
/// \param tamanoPedido tamano del array de pedidos a recorrer
/// \param listaEmpresas array de las empresas necesario para mostrarlas
/// \param tamanoEmpresas tamnio del array de empresas a mostrar
/// \param listaLocalidades array de las localidades necesario para mostrar los datos de la empresa
de forma completa
/// \param tamanoLocalidades tamano del array de localidades necesaria para recorrer
/// \return retorna -1 si las listas son nulas o 0 si se logro mostrar los clientes y la cantidad de pedidos
pendientes

```

```
int ImprimirClientesConPedidosPendientes(ePedido listaPedido[], int tamañoPedido,
eEmpresa listaEmpresas[], int tamañoEmpresas, eLocalidad listaLocalidades[], int
tamañoLocalidades);
```

## Imprimir Pedidos Pendientes

## Imprimir Pedidos Completados

```
/// \fn int ImprimirPedidos(ePedido[], int, eEmpresa[], int, int)
/// \brief Muestra la lista de empresas con pedidos en un estado en particular enviado por parametro
/// \param listaPedido array de pedidos de los cuales se verificara el estado
/// \param tamañoPedido tamaño del array de pedidos
/// \param lista lista array de empresas a mostrar
/// \param tamaño tamaño de array de empresas necesario para recorrer y mostrar
/// \param estado estado en el que se desea que esten los pedidos de los clientes a mostrar
/// \return retorna -1 si as listas son nulas o 0 si se logro mostrar correctamente

int ImprimirPedidos(ePedido listaPedido[], int tamañoPedido, eEmpresa lista[], int tamaño, int
estado);
```

## Cantidad de pedidos pendientes por locildad

```
/// \fn int BuscarPedidosPendientesPorLocalidad(ePedido[], int, eEmpresa[], int, eLocalidad[], int)
/// \brief Muestra la lista de pedidos pendientes y se pide la selccion de una localidad en la cual se
contaran los pedidos pendintes que posee
/// \param listaPedido array de pedidos que se recorrera en busca de los pendientes
/// \param tamañoPedido tamaño del array de pedidos a recorrer
/// \param listaEmpresas array de empresas necesario para mostrar la lista
/// \param tamañoEmpresas tamaño del array de empreas a recorrer
/// \param listaLocalidades array de localidades que se mostrara para que se seleccione el id de la
que se desea contar los pedidos pendientes
/// \param tamañoLocalidades tamaño del array de localidades a recorrer
/// \return retorna -1 si las listas son nulas o 0 si se logro mostrar y contar

int BuscarPedidosPendientesPorLocalidad(ePedido listaPedido[], int tamañoPedido,
eEmpresa listaEmpresas[], int tamañoEmpresas, eLocalidad listaLocalidades[], int
tamañoLocalidades);
```

## Cantidad de kilos de pp promedio reciclado por cliente

```
/// \fn int ContarKilosDePP(ePedido[], int, eEmpresa[], int)
/// \brief Mostrara el promedio por cliente de kilos de pp reciclado
```

```

/// \param listaPedido array de pedidos a recorrer en busca del pedido de una empresa
/// \param tamañoPedido tamaño del array de pedidos a recorrer
/// \param listaEmpresas array de empresas a recorrer con el que el id de cada una se comparara
con el id de empresa guardado en los pedidos
/// \param tamañoEmpresas tamaño del array de empresas a recorrer
/// \return retorna -1 si las listas son nulas o 0 si se logro sacar el promedio

int ContarKilosDePP(ePedido listaPedido[], int tamañoPedido, eEmpresa listaEmpresas[], int
tamañoEmpresas);

```

## **Cliente con mas pedidos pendientes**

## **Cliente con mas pedidos completados**

```

/// \fn int ClientesConMasPedidos(ePedido[], int, eEmpresa[], int, eLocalidad[], int, int)
/// \brief Mostrara la lista de empresas con mas pedidos en el estado recibido por parametro
/// \param listaPedido array de pedidos que se recorrera en busca de pedidos en ese estado
/// \param tamañoPedido tamaño del array de pedidos a recorrer
/// \param lista array de empresas a recorrer comparando el id con el id de empresa que guarda
cada pedio y necesaria par mostrar los clientes
/// \param tamaño tamaño del array de empresas a recorrer
/// \param listaLocalidad array de localidades necesario para mostrar de forma completa la
informacion de la empresa
/// \param tamañoLocalidad tamaño del array de localidades a recorrer
/// \param estado estado en el que se desea que esten los pedidos de las empresas a mostra
/// \return retorna -1 si las listas son nulas o 0 si se logro mostrar lo requerido

int ClientesConMasPedidos(ePedido listaPedido[], int tamañoPedido, eEmpresa lista[], int
tamaño, eLocalidad listaLocalidad[], int tamañoLocalidad, int estado);

```

## **13,**

```

/// \fn int TotalDePedidosEnviados(ePedido[], int, eEmpresa[], int, eTransporte[], int, eLocalidad[],
int)
/// \brief Muestra cada una de las empresas con la cantidad de pedidos enviados que posee y el
total a pagar por los envios y el total de km a recorrer
/// \param listaPedido array de pedidos a recorrer en busca de pedidos completados y enviados
/// \param tamañoPedido tamaño del array de pedidos a recorrer
/// \param lista array de empresas que se recorrera para luego recorrer los pedidos y ver si posee
pedidos
/// \param tamaño tamaño del array de empresas a recorrer

```

/// \param listaTransportes array de transportes que se recorrera para contar los pedidos enviados y sumar los precios y km

/// \param tamanoTransportes tamano del array de transportes a recorrer

/// \param listaLocalidad array de localidades necesario para mostrar de forma completa los datos de la empresa

/// \param tamanoLocalidad tamano del array de localidades a recorrer

/// \return retorna -1 si las listas son nulas o 0 si se pudo calcular correctamente

**int TotalDePedidosEnviados(ePedido listaPedido[], int tamanoPedido, eEmpresa lista[], int tamano, eTransporte listaTransportes[], int tamanoTransportes, eLocalidad listaLocalidad[], int tamanoLocalidad);**

Enlace al video explicativo

<https://drive.google.com/drive/folders/1C6d7nsLAtutVIUKuM4NBLTPWW-n5mATL?usp=sharing>