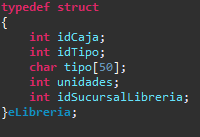
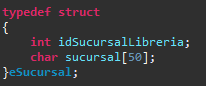
Alumno : Ian sein

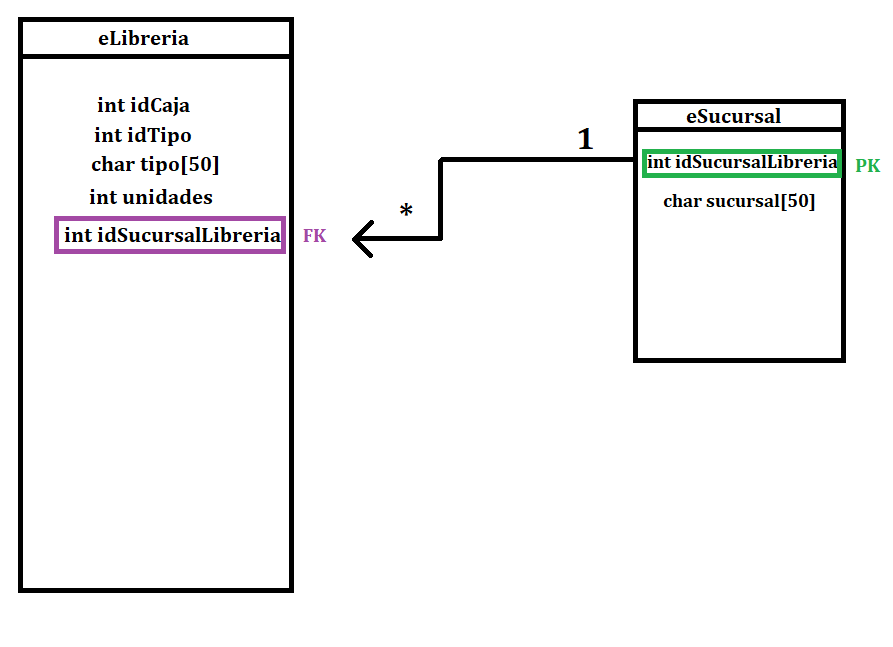
Division : 1° C

**RECUPERATORIO 2DO PARCIAL LABORATORIO**

**Estructuras**

** **

La estructura de sucursal logra relacionarse con la estructura de librería mediante el ID que esta lleva.



**Informes Anexados**

**Primer informe:** Muestra el total de unidades que se enviaron sin importar a qué sucursal hayan sido mandados los envíos.

**Segundo informe:** Muestra el total de unidades que se enviaron a la sucursal de Caballito.

**Tercer informe:** Genera y muestra un listado de los envíos quitando los que se hayan hecho hacia la sucursal de Medrano

**Cuarto informe:** Genera y muestra un listado de los envíos quitando los que se hayan hecho de libros.

**Prototipos de las funciones núcleo con su documentación**

**int parser\_EnviosFromText(char\* path , LinkedList\* listaLibreria);**

Esta función parsea los datos de los envíos desde el archivo data.csv (modo texto),

añadiendo cada elemento a la lista de librería.

**int parser\_SucursalesFromText(char\* path , LinkedList\* sucursalesLibreria);**

Esta función parsea los datos de las sucursales desde el archivo data.csv (modo texto), añadiendo cada elemento a la lista de sucursales

**int ListaEnviosLibreria(LinkedList\* listaLibreria, LinkedList\* sucursalesLibreria);**

Esta función genera un listado de todos los envíos que se hayan hecho, mostrando el tipo de unidad, cantidad de unidades y la sucursal.

**eSucursal\* ObtenerSucursalPorId(eLibreria\* auxLibreria, LinkedList\* sucursalesLibreria);**

Esta funcion devuelve una estructura de sucursal dependiendo qué estructura de librería se le haya pasado como parámetro, para poder relacionar.

**int UnidadesEnviadasPorPedido(void\* pElement);**

Esta función calcula el total de unidades que se hayan enviado en un pedido dependiendo el elemento que se le haya pasado como parámetro.

**int CalcularTotalUnidadesEnviadas(LinkedList\* listaLibreria);**

Esta función calcula el total de las unidades que hayan sido enviadas y las muestra.

**int UnidadesEnviadasACaballito(void\* pElement);**

Esta función calcula el total de unidades que hayan enviado a Caballito en un pedido dependiendo el elemento que se le haya pasado como parámetro

**int CalcularTotalUnidadesEnviadasACaballito(LinkedList\* listaLibreria);**

Esta función calcula el total de unidades que hayan sido enviadas a Caballito y las muestra.

**int FiltrarEnvioAMedrano(void\* pElement);**

Esta función valida si hay que agregar el elemento pasado como parámetro a la lista , devolviendo un número que verifique si se agrega o no. No se agregara si el elemento contiene la sucursal Medrano.

**int FiltrarTodosLosEnviosAMedrano(LinkedList\* listaLibreria, LinkedList\* sucursalesLibreria);**

Esta función muestra una lista filtrada de todos los envíos, con sus respectivos datos quitando los envíos que se hayan hecho a la sucursal de Medrano.

**int FiltrarLibro(void\* pElement)**

Esta función valida si hay que agregar el elemento pasado como parámetro a la lista , devolviendo un número que verifique si se agrega o no. No se agregara si el elemento contiene una unidad de tipo libro.

**int FiltrarTodosLosLibros(LinkedList\* listaLibreria, LinkedList\* sucursalesLibreria);**

Esta función muestra una lista filtrada de todos los envíos, con sus respectivos datos quitando los envíos que tengan como tipo de unidad los libros.

**int GuardarListaComoTexto(LinkedList\* listaLibreria, LinkedList\* sucursalesLibreria, char\* path);**

Esta función guarda un listado de todos los envíos, con sus respectivos datos quitando los envíos que se hayan hecho a la sucursal de Medrano, creando un archivo CSV en el sistema.

**eLibreria\* envio\_new()**

Esta función crea una estructura de librería de manera dinámica y la retorna.

**eLibreria\* envio\_newParametros(char\* idCajaStr, char\* idTipoStr, char\* tipoStr, char\* unidadesStr, char\* idSucursalLibreriaStr);**

Esta función Parsea todos los datos que le llegan como parámetros exceptuando los de tipo nombre, luego coloca cada dato en su respectivo campo y devuelve la estructura.

**eSucursal\* sucursal\_new()**

Esta función crea una estructura de sucursal de manera dinámica y la retorna.

**eSucursal\* sucursal\_newParametros(char\* idSucursalLibreriaStr, char\* sucursalStr);**

Esta función parsea todos los datos que le llegan como parámetros exceptuando los de tipo nombre, luego coloca cada dato en su respectivo campo y devuelve la estructura.

**int ll\_count(LinkedList\* this, int (\*fn)(void\*));**

Esta función itera una lista entera, pasándole cada elemento a la función criterio pasada como parámetro, y dependiendo lo que esta devuelva lo acumula y al final retorna el acumulador.

**LinkedList\* ll\_filter(LinkedList\* this, int(\*fn)(void\*));**

Esta función itera una lista entera, pasándole cada elemento a la función criterio pasada como parámetro, y dependiendo que retorna esta, decide si agregar el elemento a la lista o no. Luego de eso retorna la lista nueva.

**LINK A VIDEO EXPLICATIVO** <https://drive.google.com/drive/folders/16vmlR-Y-0IMOAU04tecWV_HuQlSAPvr3?usp=sharing>