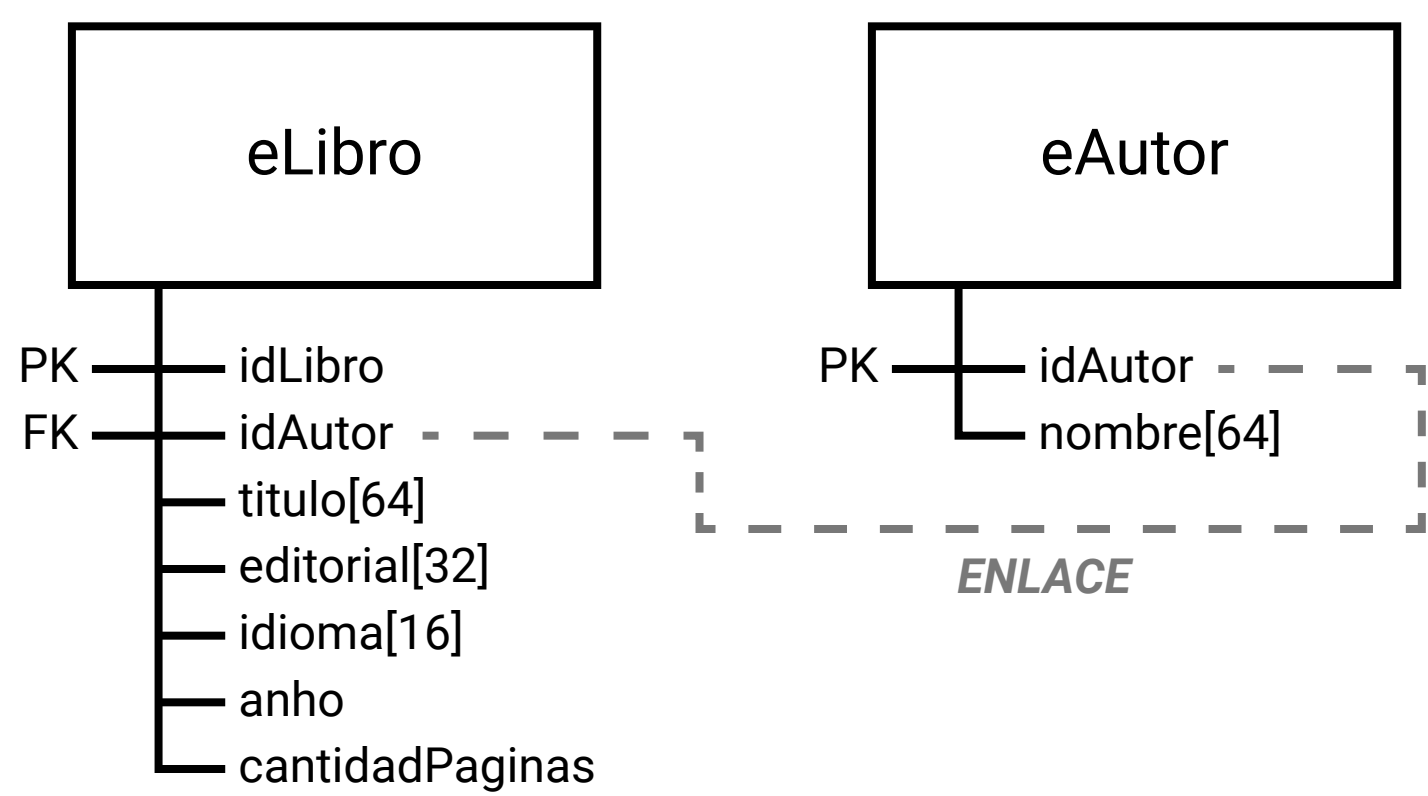


Apellido: Jacob
Nombre: Cristopher

Division: 1F
Materia: Laboratorio 1
Instancia: Recuperatorio Segundo Parcial

1. Proyecto completo en eclipse.
2. Un video que no supere los 10 minutos de duración, en donde deberán realizar la defensa del proyecto. Recuerden que deberán explicar, sin leer el código (pero si mostrándolo), las funciones más importantes del programa.
3. Un documento con:
 - a. Datos del alumno.
 - b. El detalle de la estructura agregada y su relación con las otras dos estructuras. Realizar un diagrama entidad relación.
 - c. El enunciado de los informes anexados (Punto 3 y 4).
 - d. Prototipo de cada función del núcleo del programa con la documentación correspondiente.
 - e. Enlace a drive con el video explicativo.

3) b) El programa se basa en una gestion de Libros. El mismo trabaja mediante dos listas enlazadas, cada una controlando dos tipos de datos diferentes. La principal siendo Libros y la segunda Autores. Los libros poseen una llave primaria idLibro asi como una llave foranea idAutor. El programa no permite la carga de una lista sin la otra, ya que ambas son necesarias para las funciones principales del mismo



- 3) c) Los informes creados son
- Filtrar libros de autor Stephen King
 - Filtrar libros anteriores al año 2000
 - Contar libros de autor Philip Dick
 - Contar libros que no esten en Español

4) d)

LinkedList:

Recibe el puntero de una lista y el puntero de una funcion criterio para contar resultados. Itera los nodos de la lista y los envia a la funcion criterio indicada, las devoluciones se iran acumulando en la variable contador

int ll_count(LinkedList*, int(*pCriterio)(void*));

Recibe el puntero de una lista y el puntero de una funcion criterio para filtrar resultados itera los nodos de la lista y los envia a la funcion criterio indicada, si las devoluciones son positivas, se van agregando los elementos "aprobados" a la LinkedList

LinkedList* ll_filter(LinkedList*, int (*pCriterio)(void*));

Controller:

Recibe el Puntero a una Lista y el Puntero a una funcion Parser. Pide un nombre de archivo de donde cargar los datos y de poder abrirlo, lo manda al parser indicado

int Controller_Load_From_Text(LinkedList*, char*, int(*pParser)(FILE*, LinkedList*));

Recibe el Puntero a una lista de Libros y una Lista de Autores para mostrarla unificada

int Controller_List_Books(LinkedList* pListBooks, LinkedList* pListAuthors)

Recibe el Puntero a una Lista de Libros (Origen). El puntero a una Lista Filtrada (Destino). Y el puntero a una Funcion Criterio. La lista Origen sera Filtrada con el Criterio indicado, los resultados se almacenaran en una Lista auxiliar, si dicho filtro no devuelve nulo, se cargara lo obtenido en la lista Destino

int Controller_Filter_Books(LinkedList*, LinkedList*, int(*pCriterio)(void*));

Recibe el Puntero a una Lista de Libros (Origen). El puntero a una Lista Filtrada (Destino). Y el puntero a una Funcion Criterio. La lista Origen sera Analizada con el Criterio indicado, los resultados contados se almacenaran la lista destino

int Controller_Count_Books(LinkedList*, LinkedList*, int(*pCriterio)(void*));

Recibe el Puntero a una Lista de Libros. Y el nombre del archivo que se quiere guardar. El nombre del archivo se le colocara al final una extension .csv. De cada libro se buscaran sus correspondientes datos, y se colocaran separados por comas

int Controller_Save_List_Books_As_Text(LinkedList* pListBooks, char* nombreArchivo)

Parser:

Recibe un puntero a un archivo y un puntero a LinkedList. Recorre el archivo fetcheando la data y construyendola para poder agregarla a la LinkedList

int Parser_Books_FromText(FILE*, LinkedList*);

Recibe un puntero a un archivo y un puntero a LinkedList. Recorre el archivo fetcheando la data y construyendola para poder agregarla a la LinkedList

int Parser_Authors_FromText(FILE*, LinkedList*);

Book, Author, Listing, Input:

Documentaciones de estas funciones no son consideradas parte del nucleo del programa. Incluidas en dichos archivos

3) e) Enlace Google Drive:

<https://drive.google.com/file/d/1qUH9Sg-H9ZU4F0MPd6uEMylyEAoqDGRs/view?usp=sharing>