**Descripción General BibliotecaUtils.cpp**

Este código forma parte de un sistema de gestión de biblioteca que permite manejar materiales bibliográficos (libros, revistas) y usuarios. Proporciona funcionalidades para buscar, gestionar préstamos y devoluciones, agregar materiales, gestionar usuarios y guardar o cargar información desde archivos.

**Dependencias**

* BibliotecaUtils.h: Cabecera personalizada que contiene las definiciones de las utilidades de la biblioteca.
* <fstream>: Biblioteca para manejo de archivos.
* <iostream>: Biblioteca para operaciones de entrada/salida.
* <string>: Biblioteca para manipulación de cadenas.
* <limits>: Biblioteca para gestionar límites en la entrada de datos.

**Funciones**

**bool checkIsbn(string isbn, MaterialBibliografico\* biblioteca[])**

* **Propósito:** Verifica si un ISBN es válido y si ya está registrado en la biblioteca.
* **Parámetros:**
  + isbn: ISBN que se desea validar.
  + biblioteca[]: Arreglo de punteros a objetos de tipo MaterialBibliografico, que representa el inventario de la biblioteca.
* **Retorno:**
  + true si el ISBN está vacío o ya existe en la biblioteca.
  + false si el ISBN es válido y no está registrado.

**vector<MaterialBibliografico\*> devolverCoincidencias(string dato, MaterialBibliografico\* biblioteca[])**

* **Propósito:** Realiza una búsqueda de coincidencias de nombre o autor en los materiales bibliográficos.
* **Parámetros:**
  + dato: Cadena de búsqueda, ya sea nombre o autor.
  + biblioteca[]: Arreglo de punteros a objetos de tipo MaterialBibliografico.
* **Retorno:** Vector de objetos de tipo MaterialBibliografico que coinciden con el dato de búsqueda.

**int busquedaPorId(int id, vector<User>& users)**

* **Propósito:** Busca un usuario por su ID en el vector de usuarios.
* **Parámetros:**
  + id: ID del usuario a buscar.
  + users: Vector de objetos User que representa los usuarios registrados.
* **Retorno:**
  + Índice del usuario si es encontrado.
  + -1 si el usuario no es encontrado.

**int validarNumeroBiblioteca()**

* **Propósito:** Valida que la entrada del usuario sea un número entero.
* **Retorno:** Devuelve el número ingresado si es válido.

**int opcionDeBusqueda()**

* **Propósito:** Presenta un menú para que el usuario elija si desea buscar por título o autor.
* **Retorno:** Devuelve la opción seleccionada, garantizando que esté dentro del rango permitido.

**void BibliotecaUtils::agregarMaterial()**

* **Propósito:** Permite agregar un material bibliográfico (libro o revista) a la biblioteca.
* **Detalles:**
  + El usuario ingresa los datos del material.
  + Se verifica que el ISBN sea válido y no esté repetido antes de agregar el material.
  + El material se almacena en la primera posición disponible en el arreglo de biblioteca.

**void BibliotecaUtils::mostrarInfoMateriales()**

* **Propósito:** Muestra la información de todos los materiales bibliográficos almacenados en la biblioteca.
* **Detalles:**
  + Recorre el arreglo de biblioteca y muestra la información de cada material que no sea nullptr.

**void BibliotecaUtils::buscarObj()**

* **Propósito:** Permite buscar materiales en la biblioteca por nombre o autor.
* **Detalles:**
  + Solicita un dato de búsqueda al usuario (nombre o autor).
  + Muestra las coincidencias encontradas o informa si no se encontraron resultados.

**void BibliotecaUtils::gestionUsuarios()**

* **Propósito:** Gestiona la creación, eliminación y búsqueda de usuarios en el sistema.
* **Detalles:**
  + Proporciona opciones para crear un nuevo usuario, eliminar un usuario existente o buscar un usuario por su ID.
  + En el caso de eliminación, se aseguran de que los materiales prestados sean devueltos antes de eliminar el usuario.

**void BibliotecaUtils::gestionMateriales()**

* **Propósito:** Gestiona el préstamo y la devolución de materiales bibliográficos a los usuarios.
* **Detalles:**
  + Permite al usuario realizar un préstamo seleccionando un material disponible.
  + También permite la devolución de materiales prestados a través de la identificación del usuario.

**void BibliotecaUtils::guardarInformacionBiblioteca()**

* **Propósito:** Guarda el estado actual de los materiales bibliográficos y los usuarios en archivos de texto.
* **Detalles:**
  + Los materiales se guardan en el archivo EstadoBiblioteca.txt.
  + Los usuarios y sus materiales prestados se guardan en Usuarios.txt.

**vector<string> BibliotecaUtils::split(string str, char delim)**

* **Propósito:** Divide una cadena en subcadenas utilizando un delimitador específico.
* **Parámetros:**
  + str: Cadena que se desea dividir.
  + delim: Carácter delimitador utilizado para la división.
* **Retorno:** Vector de cadenas que contiene las subcadenas resultantes.

**void BibliotecaUtils::cargarArchivoBiblioteca()**

* **Propósito:** Carga los materiales bibliográficos desde el archivo EstadoBiblioteca.txt y los añade al arreglo biblioteca.
* **Detalles:**
  + Verifica que el archivo esté correctamente abierto.
  + Divide cada línea del archivo en sus partes correspondientes (nombre, ISBN, autor, etc.) y las asigna a nuevos objetos de tipo Libro o Revista.

**void BibliotecaUtils::cargarArchivoUsuarios()**

* **Propósito:** Carga los usuarios desde el archivo Usuarios.txt y asocia sus materiales prestados en el sistema.
* **Detalles:**
  + Verifica que el archivo esté correctamente abierto.
  + Para cada usuario, asigna los materiales prestados que se encuentran también en el archivo.

**Descripción general Libro.cpp**

Este archivo implementa los métodos de la clase Libro, que es una subclase de MaterialBibliografico. La clase Libro representa materiales bibliográficos específicos, en este caso libros, y contiene información adicional como la fecha de publicación y un resumen del libro.

**Constructores y Destructores**

**Libro::Libro(string nombre, string isbn, string autor, bool estado, string fechaPublicacion, string resumen)**

* **Propósito:** Constructor de la clase Libro. Inicializa un objeto Libro con la información proporcionada, utilizando el constructor de la clase base MaterialBibliografico para inicializar los atributos comunes.
* **Parámetros:**
  + nombre: El nombre del libro.
  + isbn: El ISBN del libro.
  + autor: El autor del libro.
  + estado: El estado del libro (disponible o no).
  + fechaPublicacion: La fecha de publicación del libro.
  + resumen: El resumen del contenido del libro.

**Libro::~Libro()**

* **Propósito:** Destructor de la clase Libro. Se utiliza para liberar cualquier recurso asociado al objeto Libro. Actualmente, no realiza acciones específicas ya que la memoria dinámica no se maneja en esta clase.

**Métodos Públicos**

**string Libro::type() const**

* **Propósito:** Retorna el tipo de material bibliográfico. En este caso, siempre devuelve "libro".
* **Retorno:** Una cadena de texto con el valor "libro".

**void Libro::mostrarInformacion() const**

* **Propósito:** Muestra la información completa del libro, incluyendo la información básica proporcionada por la clase base MaterialBibliografico, junto con la fecha de publicación y el resumen.
* **Detalles:** Utiliza el método mostrarInformacion() de la clase base para mostrar los atributos comunes y luego muestra la fecha de publicación y el resumen específicos del libro.

**string Libro::get\_fechapublicacion()**

* **Propósito:** Devuelve la fecha de publicación del libro.
* **Retorno:** La fecha de publicación como una cadena de texto.

**void Libro::set\_fecha\_publicacion(string fecha\_publicacion)**

* **Propósito:** Establece la fecha de publicación del libro.
* **Parámetros:**
  + fecha\_publicacion: La nueva fecha de publicación a asignar al libro.

**string Libro::get\_resumen()**

* **Propósito:** Devuelve el resumen del libro.
* **Retorno:** El resumen como una cadena de texto.

**void Libro::set\_resumen(string resumen)**

* **Propósito:** Establece el resumen del libro.
* **Parámetros:**
  + resumen: El nuevo resumen a asignar al libro.

**Descripción General main.cpp**

Este código implementa la funcionalidad principal de un sistema de gestión de biblioteca a través de un menú interactivo. Permite al usuario agregar, mostrar, buscar materiales bibliográficos, gestionar préstamos y devoluciones, y administrar usuarios. También se manejan operaciones como la carga y el guardado de información desde y hacia archivos.

**Dependencias**

* **Librerías Estándar:**
  + <iostream>: Para realizar operaciones de entrada/salida.
  + <string>: Para el manejo de cadenas de texto.
  + <limits>: Para manejar límites de entrada, útil en la validación de datos.
* **Archivos de Cabecera Propios:**
  + MaterialBibliografico.h: Define la clase base para materiales bibliográficos.
  + BibliotecaUtils.h: Define las utilidades para gestionar la biblioteca, como agregar materiales, préstamos, devoluciones, etc.

**Funciones**

**int validarNumero()**

* **Propósito:** Valida que la entrada del usuario sea un número entero. En caso de ingresar un valor no numérico, solicita nuevamente la entrada hasta que sea válida.
* **Retorno:** El número entero ingresado por el usuario.
* **Detalles:** Utiliza cin.clear() para limpiar el estado del flujo y cin.ignore() para descartar la entrada inválida en caso de error, garantizando que el programa no se bloquee por entradas incorrectas.

**void menu()**

* **Propósito:** Despliega el menú principal del sistema de gestión de la biblioteca. Proporciona las opciones disponibles para interactuar con el sistema.
* **Detalles:**
  + Inicializa una instancia de la clase BibliotecaUtils, la cual contiene las funciones principales para gestionar los materiales y usuarios.
  + Carga los archivos de la biblioteca y los usuarios al inicio.
  + Mantiene el ciclo del menú mientras la opción seleccionada no sea 0 (salir).
  + Presenta las siguientes opciones al usuario:
    - **(1) Agregar Material:** Permite añadir un nuevo material a la biblioteca.
    - **(2) Mostrar Información:** Muestra la información de todos los materiales almacenados.
    - **(3) Buscar Material:** Permite buscar un material bibliográfico por nombre o autor.
    - **(4) Prestar y Devolver Material:** Gestiona los préstamos y devoluciones de materiales a usuarios.
    - **(5) Gestión de Usuarios:** Permite agregar, eliminar o buscar usuarios.
    - **(0) Salir:** Guarda la información de la biblioteca y finaliza el programa.
  + Después de que el usuario selecciona una opción, se llama a la función correspondiente de BibliotecaUtils.
  + Cuando el usuario selecciona 0, el programa guarda la información de la biblioteca y los usuarios en archivos y luego sale.

**int main()**

* **Propósito:** Punto de entrada principal del programa.
* **Detalles:**
  + Llama a la función menu() que gestiona el flujo del programa.
  + Retorna 0 al finalizar, indicando que el programa terminó correctamente.

**Descripción General MaterialBibliografico.cpp**

Este archivo implementa la clase MaterialBibliografico, que es la clase base para todos los materiales de la biblioteca. Esta clase contiene atributos básicos como el nombre, ISBN, autor, estado de disponibilidad, y el ID del usuario que tiene el material prestado (si corresponde). Además, proporciona métodos para obtener y modificar estos atributos, así como para mostrar la información del material.

**Constructores y Destructores**

**MaterialBibliografico::MaterialBibliografico(string nombre, string isbn, string autor, bool estado)**

* **Propósito:** Constructor de la clase MaterialBibliografico. Inicializa un objeto con los atributos básicos de un material bibliográfico.
* **Parámetros:**
  + nombre: El nombre del material.
  + isbn: El ISBN del material.
  + autor: El autor del material.
  + estado: Estado del material, donde true indica que está disponible y false que está prestado.
* **Detalles:** Inicializa el atributo idUsuario como una cadena vacía, que será usada para asociar el material a un usuario si está prestado.

**MaterialBibliografico::~MaterialBibliografico()**

* **Propósito:** Destructor de la clase MaterialBibliografico. Informa cuando un objeto de este tipo es destruido.
* **Detalles:** El destructor imprime un mensaje en consola al ser llamado.

**Métodos Públicos**

**void MaterialBibliografico::mostrarInformacion() const**

* **Propósito:** Muestra la información básica del material bibliográfico: nombre, ISBN, autor y estado (disponible o prestado).
* **Detalles:** La información se muestra a través de la salida estándar (cout).

**string MaterialBibliografico::get\_nombre()**

* **Propósito:** Devuelve el nombre del material bibliográfico.
* **Retorno:** El nombre del material como una cadena de texto.

**void MaterialBibliografico::set\_nombre(string nombre)**

* **Propósito:** Establece un nuevo nombre para el material bibliográfico.
* **Parámetros:**
  + nombre: El nuevo nombre que se asignará al material.

**string MaterialBibliografico::get\_isbn()**

* **Propósito:** Devuelve el ISBN del material bibliográfico.
* **Retorno:** El ISBN del material como una cadena de texto.

**void MaterialBibliografico::set\_isbn(string isbn)**

* **Propósito:** Establece un nuevo ISBN para el material bibliográfico.
* **Parámetros:**
  + isbn: El nuevo ISBN que se asignará al material.

**string MaterialBibliografico::get\_autor()**

* **Propósito:** Devuelve el autor del material bibliográfico.
* **Retorno:** El autor del material como una cadena de texto.

**void MaterialBibliografico::set\_autor(string autor)**

* **Propósito:** Establece un nuevo autor para el material bibliográfico.
* **Parámetros:**
  + autor: El nuevo autor que se asignará al material.

**bool MaterialBibliografico::get\_estado()**

* **Propósito:** Devuelve el estado del material (disponible o prestado).
* **Retorno:** true si el material está disponible, false si está prestado.

**void MaterialBibliografico::set\_estado(bool estado)**

* **Propósito:** Establece el estado del material bibliográfico (disponible o prestado).
* **Parámetros:**
  + estado: El nuevo estado del material (true para disponible, false para prestado).

**string MaterialBibliografico::get\_id()**

* **Propósito:** Devuelve el ID del usuario que tiene el material prestado.
* **Retorno:** El ID del usuario como una cadena de texto.

**void MaterialBibliografico::set\_id(string idUsuario)**

* **Propósito:** Asigna un ID de usuario al material bibliográfico, para indicar quién tiene el material prestado.
* **Parámetros:**
  + idUsuario: El ID del usuario que tiene el material prestado.

**Descripción General Revista.cpp**

Este archivo implementa la clase Revista, que es una subclase de MaterialBibliografico. La clase Revista representa revistas dentro del sistema de la biblioteca y contiene atributos adicionales específicos de este tipo de material, como el número de edición y el mes de publicación.

**Constructores y Destructores**

**Revista::Revista(string nombre, string isbn, string autor, bool estado, int numEdicion, int mesPublicacion)**

* **Propósito:** Constructor de la clase Revista. Inicializa un objeto Revista con los atributos proporcionados, utilizando el constructor de la clase base MaterialBibliografico para inicializar los atributos comunes.
* **Parámetros:**
  + nombre: El nombre de la revista.
  + isbn: El ISBN de la revista.
  + autor: El autor de la revista.
  + estado: El estado de la revista, true si está disponible y false si está prestada.
  + numEdicion: El número de edición de la revista.
  + mesPublicacion: El mes de publicación de la revista.

**Revista::~Revista()**

* **Propósito:** Destructor de la clase Revista. Se llama cuando un objeto Revista es destruido.
* **Detalles:** El destructor imprime un mensaje en la consola al ser llamado para indicar que el objeto Revista está siendo eliminado.

**Métodos Públicos**

**string Revista::type() const**

* **Propósito:** Devuelve el tipo de material bibliográfico, que en este caso es "revista".
* **Retorno:** Una cadena de texto con el valor "revista".

**void Revista::mostrarInformacion() const**

* **Propósito:** Muestra toda la información relevante de la revista, incluyendo la información básica de la clase MaterialBibliografico y los detalles específicos de la revista (número de edición y mes de publicación).
* **Detalles:** Llama al método mostrarInformacion() de la clase base para mostrar la información general, y luego muestra los atributos específicos de la revista.

**int Revista::get\_numEdicion()**

* **Propósito:** Devuelve el número de edición de la revista.
* **Retorno:** El número de edición como un entero.

**void Revista::set\_numEdicion(int numEdicion)**

* **Propósito:** Establece un nuevo valor para el número de edición de la revista.
* **Parámetros:**
  + numEdicion: El nuevo número de edición que se asignará.

**int Revista::get\_mesPublicacion()**

* **Propósito:** Devuelve el mes de publicación de la revista.
* **Retorno:** El mes de publicación como un entero.

**void Revista::set\_mesPublicacion(int mesPublicacion)**

* **Propósito:** Establece un nuevo mes de publicación para la revista.
* **Parámetros:**
  + mesPublicacion: El nuevo mes de publicación que se asignará.

**Descripción General User.cpp**

Este archivo implementa la clase User, que representa a los usuarios de la biblioteca. Los usuarios pueden prestar y devolver materiales bibliográficos, así como visualizar los materiales que tienen actualmente prestados. La clase también maneja la lógica de limitar a los usuarios a un máximo de 5 materiales prestados simultáneamente.

**Constructores y Destructores**

**User::User(string nombre, int id)**

* **Propósito:** Constructor de la clase User. Inicializa un objeto User con un nombre y un ID.
* **Parámetros:**
  + nombre: Nombre del usuario.
  + id: Identificador único del usuario.

**User::~User()**

* **Propósito:** Destructor de la clase User. Se llama cuando el objeto User es destruido. Actualmente, no realiza ninguna operación específica.

**Métodos Públicos**

**void User::prestarMaterial(MaterialBibliografico \*material)**

* **Propósito:** Permite que el usuario preste un material bibliográfico si está disponible.
* **Parámetros:**
  + material: Puntero a un objeto MaterialBibliografico que representa el material que se desea prestar.
* **Detalles:**
  + Verifica si el material está disponible para prestar (si su estado es true).
  + Si el usuario tiene menos de 5 materiales prestados, el material se asocia al usuario, cambia su estado a no disponible, y se registra el ID del usuario en el material.
  + Si el usuario ya tiene 5 materiales prestados, muestra un mensaje indicando que no puede prestar más.

**void User::devolverMaterial()**

* **Propósito:** Permite al usuario devolver un material que tiene prestado.
* **Detalles:**
  + Muestra todos los materiales prestados del usuario.
  + Solicita al usuario que ingrese el ISBN del material que desea devolver.
  + Si el ISBN es correcto, se cambia el estado del material a disponible, se elimina el ID del usuario asociado y se libera el espacio en el arreglo de materiales prestados.

**void User::devolverTodosLosMateriales()**

* **Propósito:** Permite devolver todos los materiales prestados por el usuario.
* **Detalles:**
  + Recorre el arreglo de materialesPrestados y marca todos los materiales como disponibles, eliminando la asociación con el usuario.

**void User::mostrarMaterialesPrestados()**

* **Propósito:** Muestra la información de todos los materiales que el usuario tiene actualmente prestados.
* **Detalles:**
  + Si el usuario no tiene ningún material prestado, muestra un mensaje indicando que no posee materiales.
  + De lo contrario, muestra la información de cada material prestado.

**Métodos Getter y Setter**

**string &User::get\_nombre()**

* **Propósito:** Devuelve el nombre del usuario.
* **Retorno:** Referencia a la cadena de texto que contiene el nombre del usuario.

**void User::set\_nombre(const string &nombre)**

* **Propósito:** Establece un nuevo nombre para el usuario.
* **Parámetros:**
  + nombre: El nuevo nombre del usuario.

**int User::get\_id()**

* **Propósito:** Devuelve el ID del usuario.
* **Retorno:** El ID del usuario como un entero.

**void User::set\_id(int id)**

* **Propósito:** Establece un nuevo ID para el usuario.
* **Parámetros:**
  + id: El nuevo ID del usuario.

**MaterialBibliografico \*\*User::get\_materialePrestados()**

* **Propósito:** Devuelve el arreglo de materiales prestados por el usuario.
* **Retorno:** Un puntero a un arreglo de punteros de MaterialBibliografico, que contiene los materiales que el usuario tiene prestados.