

Algoritmos y Estructuras de Datos

Entrega final

1. Título, integrantes y repositorio

- a. Biblioteca inteligente, Sistema de Gestión de Biblioteca.
- b. Juan Camilo Pugliese Pomares.

2. Resumen ejecutivo:

El problema identificado fue la falta de un sistema eficiente para gestionar los datos en una biblioteca. Esta carencia llevaba a dificultades para encontrar y organizar los libros, así como para controlar el préstamo y devolución de los mismos. Para abordar esta situación, se propuso desarrollar un código que implementara un sistema de gestión de datos bibliotecarios.

La solución propuesta consistió en crear dos clases principales: `Book` y `Library`. La clase `Book` representaba un libro y contenía información como título, autor, año de publicación, categoría y disponibilidad. Por su parte, la clase `Library` permitía agregar libros, buscar por diferentes criterios (título, autor, categoría, año), obtener libros disponibles, tomar prestado y devolver libros.

El código implementado ofreció una solución eficaz al problema planteado. Permitió una gestión más eficiente de los libros en la biblioteca, facilitando la búsqueda, organización y control de préstamos. Además, se proporcionaron funcionalidades adicionales como la ordenación de libros por año y la impresión de los detalles de estos.

En conclusión, el código desarrollado solucionó la necesidad de un sistema de gestión de datos en una biblioteca, mejorando significativamente su eficiencia y proporcionando una herramienta útil para el manejo de libros y préstamos.

3. Funcionalidad de la solución computacional:

La solución computacional propuesta, mediante el código descrito, ofrece las siguientes funcionalidades para abordar el problema de la gestión de datos en una biblioteca:

- a. **Agregar libros a la biblioteca:** El código permite agregar nuevos libros a la colección de la biblioteca.
- b. **Buscar libros por diferentes criterios:** Es posible buscar libros por título, autor, categoría y año de publicación. Esto facilita la localización de libros específicos en la biblioteca.
- c. **Obtener libros disponibles:** La solución permite obtener una lista de libros que están disponibles para préstamo.

- d. **Tomar prestado y devolver libros:** El sistema permite el préstamo y la devolución de libros, actualizando automáticamente su disponibilidad y manteniendo registros de los prestatarios.
- e. **Ordenar libros por año de publicación:** Existe una función para ordenar los libros según su año de publicación, lo cual ayuda a organizar la biblioteca y facilita la búsqueda de libros históricos o recientes.
- f. **Imprimir detalles de los libros:** La solución proporciona una función para imprimir los detalles de los libros, lo que permite visualizar de manera clara la información relevante de cada libro.
- g. **En resumen,** la solución computacional ofrece una forma eficiente de agregar, buscar, prestar, devolver y organizar libros en una biblioteca. Esto ayuda a agilizar la gestión de datos y a mejorar el control sobre la disponibilidad de los libros, facilitando el acceso a la información bibliográfica y mejorando la experiencia de los usuarios de la biblioteca.

4. Objetivos específicos:

- a. Se cumplió el objetivo en concreto.
- b. Se cumplió el objetivo en concreto.
- c. Se cumplió el objetivo en concreto.
- d. Se decidió no implementar ningún algoritmo de búsqueda.
- e. Se implementó el sistema de préstamo y devolución de libros.

5. Herramientas de programación:

- a. Clases
- b. Constructores
- c. Métodos
- d. Funciones
- e. Ciclos for
- f. Ciclos if
- g. Apuntadores

6. Descripción del programa:

- a. **Nombres de los archivos que componen el programa**
 - i **proyecto_final.cpp**
 - ii **proyecto_final.exe**
 - iii **index.html**
- b. Qué archivo contiene el bloque de ejecución
 - i **proyecto_final.exe**
- c. Que clases y/o funciones se definen en cada archivo

El código proporcionado contiene las siguientes clases:

- i **Book:** Esta clase representa un libro en la biblioteca y tiene las propiedades explicadas arriba.

- ii Library: Esta clase representa una biblioteca y tiene las siguientes funcionalidades:
- iii books: Un vector que almacena los libros en la biblioteca.
- iv addBook: Agrega un libro a la biblioteca.
- v findBook: Busca un libro por su título y devuelve un puntero al libro encontrado.
- vi searchByAuthor: Busca libros por autor y devuelve un vector de punteros a los libros encontrados.
- vii searchByCategory: Busca libros por categoría y devuelve un vector de punteros a los libros encontrados.
- viii searchByYear: Busca libros por año de publicación y devuelve un vector de punteros a los libros encontrados.
- ix getAvailableBooks: Obtiene todos los libros disponibles en la biblioteca y devuelve un vector de punteros a los libros.
- x borrowBook: Presta un libro a un prestatario si está disponible.
- xi returnBook: Devuelve un libro prestado a la biblioteca.

Además, el código también incluye las siguientes funciones globales:

- xii sortByYear: Ordena un vector de libros por año de publicación.
- xiii printBooks: Imprime la información de los libros en un vector dado.
- xiv La función main es el punto de entrada del programa y se utiliza para probar las funcionalidades de la clase Library, como agregar libros, buscar libros por diferentes criterios, prestar libros y devolver libros.

d. Flujo del programa

- i El programa comienza definiendo la clase Book que representa un libro en la biblioteca y tiene propiedades como título, autor, año, categoría, disponibilidad y prestatario. Luego se define la clase Library, que representa la biblioteca y contiene una colección de libros y funciones para agregar libros, buscarlos por diferentes criterios, prestar y devolver libros. Después se definen dos funciones globales, sortByYear para ordenar los libros por año y printBooks para imprimir la información de los libros. En la función main, se crea una instancia de la clase Library llamada library. Se crean varios objetos de la clase Book con diferentes títulos, autores, años y categorías, y se agregan a la biblioteca utilizando el método addBook de la instancia library. A continuación, se realizan diversas operaciones, como buscar libros por título, autor, categoría y año utilizando los métodos de búsqueda de la biblioteca. Luego se obtienen los libros disponibles utilizando el método getAvailableBooks y se imprimen sus detalles utilizando la función printBooks. A continuación, se prueba el préstamo de un libro utilizando el método borrowBook, verificando si el libro está disponible y mostrando un mensaje correspondiente. Finalmente, se prueba la devolución de un libro utilizando el método returnBook, comprobando si el libro estaba prestado por el prestatario indicado y mostrando mensajes apropiados. En resumen, el flujo del programa implica la creación de la biblioteca, la adición de libros, la búsqueda y visualización de libros, el préstamo y la devolución de libros, todo ello utilizando

los métodos y funciones definidos en las clases y funciones respectivas. Por último, se implementó un `std::cin.ignore()` debido a que se cerraba de inmediato al ejecutar, con esto logramos que el usuario decida cuando cerrar el archivo ejecutado.