

Algoritmos y Estructuras de Datos I

**PROYECTO INDIVIDUAL – PARTE I**

Francisco Piria

Contenido

[Introducción 3](#_Toc417411916)

[Problema planteado 3](#_Toc417411917)

[Análisis de alternativas 3](#_Toc417411918)

[Algoritmos 3](#_Toc417411919)

[Selección y justificación de alternativa a implementar 3](#_Toc417411920)

[Conclusiones 3](#_Toc417411921)

[Guía del usuario 3](#_Toc417411922)

# Introducción

El trabajo realizado consiste de una aplicación Java de consola para la empresa UcuBooks, que permite gestionar una biblioteca virtual de libros. El principal uso de esta aplicación es permitirle a los clientes realizar búsquedas de libros por diferentes criterios, entre otros.

## Problema planteado

Las tareas que debe poder realizar la aplicación incluyen:

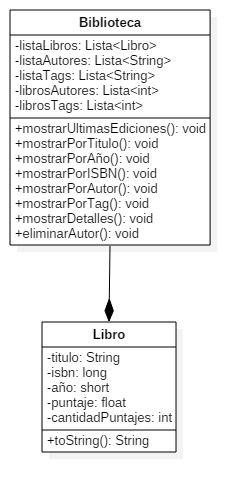
* Permitir hacer búsquedas de libros por nombre, autor, genero, año o ISBN y mostrar en pantalla los resultados
* Mostrar en pantalla los detalles de un libro en particular
* Mostrar las últimas ediciones a partir de un cierto año
* Eliminar un autor y todos sus libros

Todas estas funcionalidades se deben poder realizar a través de líneas de comando.

# Análisis de alternativas

A continuación se describirán dos alternativas para solucionar las tareas que exige UcuBooks, las cuales implementan el tipo de datos abstracto Lista.

Primera alternativa

La primera alternativa tiene la siguiente estructura:

La clase Biblioteca, que sera la encargada de realizar las tareas de UcuBooks, contiene 5 listas. Cada una de esas listas se corresponde con uno de los archivos de la base de datos de UcuBooks. Una de las listas contiene objetos Libro y las etiquetas de sus nodos son los ID de los libros. Otra lista contiene strings que representan a los autores (con etiqueta ID para cada autor). Otra hace lo mismo para los tags que puede tener un libro, y las ultimas dos establecen una conexione entre IDs de libros y ID de autores y tags respectivamente.

### Algoritmos

* Descripción en seudocódigo (de acuerdo a los estándares de la asignatura) de las diferentes funcionalidades que el programa debe cumplir, para cada alternativa de representación indicada arriba.
* Análisis del orden del tiempo de ejecución de cada funcionalidad, para cada alternativa de representación indicada arriba.

# Selección y justificación de alternativa a implementar

Breve fundamentación de la elección.

Descripción detallada de la alternativa implementada (dibujos, diagramas, aclaraciones sobre clases e interfaces).

# Conclusiones

Resumen de las características más relevantes del producto. ¿Por qué debería el cliente elegir mi solución?

# Guía del usuario

* Instrucciones (breves) para el uso de la aplicación.
* Consideraciones de instalación (si corresponde) y requerimientos no funcionales para la operación.