

## PROYECTO INDIVIDUAL – PARTE 1 - Propuesta inicial

### Objetivo

Aplicar los conocimientos adquiridos durante el curso en la realización de una aplicación real, analizando diferentes alternativas.

### Escenario

La empresa **UcuBooks**, gestora global de una biblioteca virtual de libros, desea brindar a sus clientes una app que les permita buscar libros por diferentes criterios, a la vez que generar listas de libros leídos y otras funcionalidades típicas de este tipo de aplicaciones.

Para ello se ha de desarrollar una aplicación de una aplicación **Java de consola (sin interfaz gráfica, solo mediante línea de comando)** que realice las principales tareas de **UcuBooks**, por ejemplo, entre otras:

- Mostrar las últimas ediciones (a partir de cierta fecha)
- Mostrar información detallada de un libro
- Permitir hacer búsquedas de libros por nombre y año, búsquedas de autores e ISBN
- Mostrar los libros escritos por determinado autor
- Mostrar los libros de cierto género
- Eliminar un autor (recibido por parámetro) y eliminar todos sus libros asociados.

### Trabajo a ser desarrollado

Se debe realizar una propuesta de producto solicitado por la empresa **UcuBooks**. Esta propuesta ha de contener:

- El programa solicitado, cumpliendo completa y correctamente con todas las funcionalidades solicitadas, con todos los accesorios necesarios para su ejecución.
- Todo el código fuente correspondiente
- Los tests unitarios de toda la lógica interna del programa
- La Documentación Javadoc completa del programa desarrollado, con el objetivo de facilitar luego la posterior extensión del mismo, así como la reutilización de la lógica en otros proyectos de la compañía
- Un informe de análisis de alternativas de implementación (al menos dos combinaciones diferentes de estructuras de datos) y justificación de la elección de la alternativa implementada. Este informe deberá incluir una evaluación de los órdenes del tiempo de ejecución de las distintas operaciones implementadas, evaluación de los consumos de memoria involucrados y consideraciones referentes a la escalabilidad del sistema desarrollado.

## Funcionalidades requeridas

El programa:

- Leerá por consola los archivos de datos iniciales y los archivos que contendrán las operaciones a realizar.
- Permitirá realizar por consola las consultas
- Emitirá por consola los resultados de dichas consultas.

La implementación debe estar basada en el **TDA LISTA** desarrollado en la Unidad Temática 3. **(NO SE ACEPTAN IMPLEMENTACIONES QUE UTILICEN NINGÚN OTRO TIPO DE COLECCIONES O ESTRUCTURAS)**

## Presentación y evaluación

Esta primera parte del Proyecto Individual deberá ser entregada en webasignatura:

- El lunes **30 de abril**: documento Word el informe de análisis de alternativas
- Lunes **21 de mayo**: Un archivo *Zip* con todo el código fuente del programa desarrollado, incluyendo archivos de prueba y casos de test, el informe definitivo y una presentación formato Power Point para la defensa
- El trabajo será presentado públicamente entre el día **24 de mayo**, a partir de la hora **18:15**. La presentación tendrá una duración máxima de 5 minutos, más 2 minutos para preguntas y sugerencias de pares y Profesores.

El Proyecto Individual será evaluado, pudiendo tener las siguientes calificaciones:

- Insuficiente – El software entregado no cumple mínimamente con todos los requerimientos indicados, y/o el informe no es lo suficientemente completo y exhaustivo, o la presentación / defensa pública no es satisfactoria. Se deberán enmendar los errores indicados no después del día **4 de abril**.
- Suficiente – El Proyecto desarrollado satisface mínimamente las condiciones indicadas en todos los componentes
- Destacado – Tanto el software, como el informe y la presentación exceden significativamente las condiciones mínimas requeridas.

## ANEXO I: Estructura de la base de datos provista:

La base de datos se encuentra distribuida en **8 archivos de texto** (con campos separados por “pipe” – “|”):

1. Libros.txt
2. Autores.txt
3. LibrosAutores.txt
4. Tags.txt
5. LibrosTags.txt

1: “**Libros.txt**”. Archivo con todo el detalle de los libros. Campos:

- a) **id** - numérico entero
- b) **Nombre** - String
- c) **año** - numérico entero (no hay restricciones, pero van a ser siempre de 4 dígitos)
- d) **puntaje** - numérico con decimal (entre 0.00 y 5.00)
- e) **cantidadPuntajes** – numérico entero
- f) **ISBN/ISBN13** separados con “/”. String

Ejemplo de registro de un libro:

**1|The Hunger Games (The Hunger Games, #1)|2008|4.34|4780653|439023483|9780439023480**

2: “**Autores.txt**”. Archivo con los autores. Campos:

- a) **id** - numérico entero
- b) **Nombre** - String

Ejemplo de registro de actor:

**1|Suzanne Collins**

3: “**LibrosAutores.txt**” Archivo que contiene la asociación entre un libro y su(s) autor(es):

- a) **identificador del libro**- número entero
- b) **identificador de autor** - número entero.

**1|1**

4: “**Tags.txt**”. Archivo con los tags que puede tener un libro. Campos:

- a) **id** - numérico entero
- b) **nombreTag** – String

**908|2017\_horror**

4: **"LibrosTags.txt"**. Archivo con los tags que tiene un libro. Campos:

- a) **idLibro** - numérico entero
- b) **idTag** – numérico entero