# Store “Léame de una”

***“Léame de una”*** es una empresa dedicada a la comercialización de libros. En esta primera versión de la aplicación se requiere que el software gestione información de libros, editoriales y autores. Debe permitir la creación de libros a través de un formulario, debe presentar un listado de libros, debe presentar un listado de editoriales y un listado de autores. Se deben usar mínimo tres persistencias en formato texto.

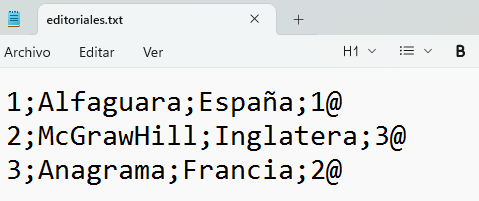
Para esta primera versión del programa se definen tres clases: Editorial, Autor y Libro. Está prohibido cambiar los nombres de las clases, los tipos de datos de las propiedades y su orden. **Solo se permiten constructores vacíos en las tres clases**. A continuación, se presentan las clases y sus propiedades:

|  |  |  |
| --- | --- | --- |
|  |  |  |

## Clase Editorial

La clase **Editorial** contiene el id, el nombre, el país y un formato que puede ser: 1 (impreso), 2 (digital) y 3 (impreso y digital), la propiedad cantidad de libros sirve para saber cuántos libros tiene la editorial.

La persistencia para almacenar la editorial debe llamarse “***editoriales.txt”***. A continuación, se presenta un ejemplo de la persistencia que almacena las editoriales.



Los datos en “***editoriales.txt”***, se deben mostrar en el siguiente orden:

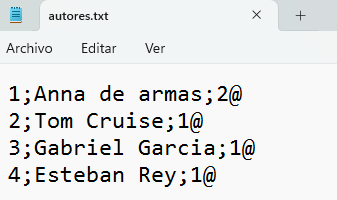
idEditorial(la cual debe ser un numero generado automáticamente por el software);nombreEditorial; PaisEditorial;formatoEditorial.

Observe que la cantidad de libros de la editorial **NO** se almacena en la persistencia. La cantidad de libros de la editorial depende de los libros existentes que hayan sido agregados por medio del formulario que usted implementará posteriormente.

## Clase Autor

La clase **Autor** contiene el id, el nombre, el género, la propiedad cantidad de libros sirve para saber cuántos libros tiene el autor.

La persistencia para almacenar los autores debe llamarse “***autores.txt***”. A continuación, se presenta un ejemplo de la persistencia que almacena los autores.



Los datos en “***autores.txt***”, se deben mostrar en el siguiente orden:

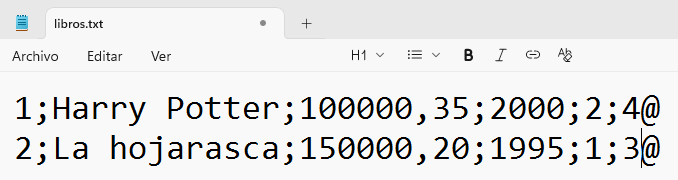
idAutor(el cual debe ser un numero generado automáticamente por el software);nombreAutor;generoAutor.

Observe que la cantidad de libros de cada autor **NO** se almacena en la persistencia. La cantidad de libros del autor depende de los libros existentes que hayan sido agregados por medio del formulario que usted implementará posteriormente.

## Clase Libro

La clase **Libro** contiene el id, el nombre, el precio, año, una Editorial y un Autor.

La persistencia para almacenar los libros debe llamarse “***libros.txt***”. A continuación, se presenta un ejemplo de la persistencia que almacena los autores.



Los datos en ***libros.txt***”, se deben mostrar en el siguiente orden:

idLibro (el cual debe ser un numero generado automáticamente por el software); nombreLibro;precioLibro((NO se acepta formato exponencial, es decir, manteniendo el tipo de dato **Double** pero mostrando el numero completo junto a sus decimales);anioLibro;idEditorial;idAutor

## Consideraciones de calidad

Todo el software de la Tienda ***“Léame de una”*** *debe* ser desarrollado bajo estrictos estándares de código que garanticen calidad, seguridad y mantenimiento eficiente. La arquitectura del sistema sigue patrones de diseño sólidos, asegurando una estructura modular y adaptable a futuras mejoras. Uno de los pilares fundamentales en su construcción es el modelo Vista-Controlador, el cual separa la lógica de negocio de la interfaz gráfica, facilitando el desarrollo y mantenimiento del software de manera ordenada y escalable.

Además, todas las aplicaciones de ***“Léame de una”*** se han implementado utilizando el paradigma de programación orientada a objetos, lo que permite reutilización de código, encapsulamiento y una organización clara de las funcionalidades del sistema. Con esta metodología, se espera que el software ofrezca una experiencia fluida y optimizada para los usuarios, asegurando estabilidad y eficiencia en cada interacción.

La empresa ha optado por utilizar JavaFX en lugar de Swing y AWT debido a su superioridad en términos de rendimiento, escalabilidad y modernidad. A diferencia de Swing, que depende de componentes más antiguos y de una estructura menos flexible, JavaFX ofrece una arquitectura moderna basada en hardware acelerado, lo que permite aplicaciones más fluidas y con un diseño atractivo. Además, JavaFX facilita la integración de efectos visuales avanzados, animaciones y gráficos sofisticados sin comprometer el rendimiento, asegurando una experiencia más intuitiva para los usuarios.

Otro factor determinante para la selección de la tecnología utilizada en los desarrollos de la tienda ***“Léame de una”***, es la capacidad que tiene JavaFX para adaptarse a múltiples plataformas y su compatibilidad con tecnologías modernas como CSS y FXML, las cuales simplifican el desarrollo de interfaces visuales. Mientras que AWT tiene limitaciones en cuanto a personalización y adaptación a interfaces contemporáneas, JavaFX proporciona herramientas robustas para la construcción de UI modernas con diseños responsivos y dinámicos.

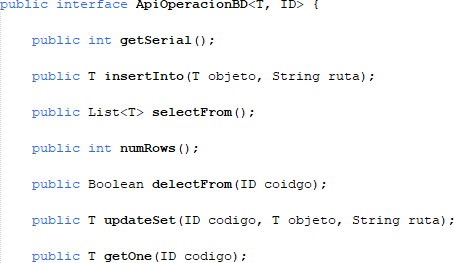
***“Léame de una”*** ha decidido apostar por esta tecnología porque garantiza estabilidad, mantenimiento eficiente y una evolución acorde a los estándares actuales de desarrollo de software.

# ¿Qué espera el cliente?

1. Que la aplicación ***compile correctamente***, debe tener todas las librerías necesarias para su correcto funcionamiento.
2. Se requiere construir un formulario que permita la creación de libros con todas sus propiedades excepto el idLibro (idLibro debe ser consecutivo y generado automáticamente por el software). Para seleccionar la editorial o el autor, el desarrollador debe utilizar combos, de tal manera que la información de las editoriales y los autores se obtenga desde dos persistencias.
3. Se debe construir una tabla que permita visualizar la información de los libros: Código del libro, nombre del libro, precio del libro (NO se acepta formato exponencial, es decir, manteniendo el tipo de dato **Double** pero mostrando el numero completo junto a sus decimales), el año del libro, el nombre de la editorial y entre paréntesis el nombre del país de esa editorial, también el nombre del autor y entre paréntesis el género del autor, no se acepta true o false, debe imprimir masculino si es true y femenino si es false.
4. El formulario y las tablas deben ser responsivos
5. Se debe construir una tabla que permita visualizar la información de las editoriales: Código, nombre, país, formato y la cantidad de libros (depende de la cantidad de libros registrados en esa editorial).
6. Se debe construir una tabla que permita visualizar la información de los autores: Código, nombre, género y la cantidad de libros (depende de la cantidad de libros registrados para los autores).

## LEAME.TXT

Los servicios deben implementar la siguiente interfaz.



Si la solución no implementa el modelo vista controlador en conjunción con buenas prácticas de desarrollo se anularán los puntos.

Por favor NO haga formularios o listados que NO se solicitan, enfoque sus esfuerzos en implementar solo lo que está definido en el apartado *“¿Qué espera el cliente?*”

Dentro del repositorio de github se encuentra la carpeta **“Guia”** dentro de esta carpeta se encuentran las clases de otro proyecto que debe ser utilizado como guía para crear e implementar los requisitos del programa. La sintaxis y diseño del código se debe implementar de la misma manera que en el proyecto **“Guia”**