



# Unab Online

## ***Biblioteca Universidad Andrés Bello***

***Profesionales en formación: Esteban Olivares***

***Javier Sanhuesa***

***Bastían Paredes***

***Eduardo Ureta***

**Docentes: Matías Vargas, Álvaro Trehwela**

## **INTRODUCCIÓN**

El proyecto realizado tiene por objetivo automatizar los procesos necesarios para prestar libros en la biblioteca de una universidad ficticia. El programa permite extraer la información de una base de datos, establecida localmente en archivos formato .txt, para mostrar todos los usuarios y libros registrados. También permite crear, leer, modificar y borrar información de los ficheros.

Si bien Java requiere inglés para programar, las variables, funciones y clases fueron escritas en español para un mejor entendimiento de quien lea el código. Las tildes de las variables y funciones no se escribieron.

## DIAGRAMA DE CLASES



Se consideró realizar una herencia de la clase *Usuario*, ya que contiene todos los atributos y métodos que comparten *Estudiante* y *Docente*, excepto por los métodos `rawData()` y `OrderedData()` (que inician en *Usuario*), y les permite modificar información en relación al guardado de los ficheros. Además de esto, le permite tener un código más limpio y fácil de entender al momento de revisar los atributos y sus relaciones.

Todos los atributos se realizaron en estado `private`, ya que serán accesibles solo desde getters y setters públicos con la finalidad de mantener la información contenida directamente conectada con los desarrolladores y no con el usuario. Cada atributo tiene su propio getter y setter, donde es en este último que se comprueba que los atributos cumplan con las condiciones solicitadas.

Adicional a esto, la cardinalidad que presentan los atributos es de 0 a 1 (en caso de los Estudiantes y Docentes), limitando la cantidad de libros que puede tener cada estudiante/docente y viceversa. Así teniendo un registro que condiciona la cantidad de libros que pueden estar dentro de la variable libro, considerando el uso de `rawData()`.

Se agregó el método `rawData()` en todas las clases para poder guardar su información en los ficheros. Además se agregó el método `orderedData()` para mostrar la información de manera ordenada al usuario cuando lo solicite.

## **Conclusiones**

El programa cumple con su función de administrar los libros y usuarios de la biblioteca de manera efectiva. Los retos que se tuvieron durante el desarrollo se presentaron al responder las preguntas ¿cómo establezco una base de datos? ¿cómo java lee y modifica archivos? ¿cómo cargo los datos de la base de datos en la RAM en forma de objetos? ¿cómo evito que los datos erróneos ingresados por el usuario generen un error que detenga la ejecución? Adicional a esto, también queda pendiente la realización de una interfaz gráfica que pueda pedir datos al usuario y la profundización en algo que sea más intuitivo para el usuario, más que solo una interacción de consola directamente.