



Trabajo final. Anteproyecto

Fecha entrega: 25/04/2017

Autor: Victoriano Sevillano Vega

Trabajo final. Anteproyecto.

Biblioteca en Java

En este proyecto vamos a gestionar las diferentes publicaciones de una biblioteca, en la cual podremos encontrar diferentes revistas, periódicos, novelas y libros de texto.

Nuestra biblioteca podrá ser gestionada de forma se puedan añadir, eliminar, listar o buscar dichas publicaciones.

Además, en nuestra biblioteca se podrán prestar y devolver todas las publicaciones, teniendo estos diferente tiempo de préstamo en función de diferentes factores en la publicación a prestar.

También podremos realizar las opciones de listar solamente aquellos libros que estén prestados, así como listar solo los libros a devolver hoy.

1. ArrayList (estructura de datos)

La estructura de datos que va a contener este proyecto, va a ser una estructura de libros, de diferentes tipos (novelas, revistas, periódicos, libros de texto, etc.)

Con esta estructura de libros se podrán realizar varias funciones:

- **Añadir publicación → novela, revista, periódico, libro de texto.**
- **Borrar publicación (por índice o identificador)**
- **Listar publicaciones.**
- **Listar novelas, periódicos, revistas o libros de texto.**
- **Buscar publicación por título o identificador.**

Todos los libros se pueden prestar, para ello además, implementaremos otro menú donde podamos:

- **Prestar libro**
- **Devolver libro**

- **Listar prestados**
- **Libros a devolver hoy**

2. Enumeraciones

Habr  una enumeraci n para las diferentes tem ticas
(Enum GeneroNovela):

- **Policial**
- **Rom ntica**
- **Aventura**
- **Terror**
- **Ciencia Ficci n**
- **Investigaci n**

Enumeraci n para el g nero de los peri dicos

(Enu GeneroPeriodico):

- **Pol tica**
- **Sociedad**
- **Deportes**
- **Otros**

Enumeraci n para el g nero de los peri dicos

(Enu GeneroRevista):

- **Pol tica**
- **Econom a**
- **Sociedad**
- **Deportes**

- **Cultura**
- **Varios**

Habr  una enumeraci n para el periodo de publicaci n de revistas y peri dicos (Enum Periodo):

- **Diaria**
- **Semanal**
- **Mensual**
- **Anual**

Habr  una enumeraci n para la tem tica de los libros de texto (Enum Materia):

- **Lenguaje**
- **Matem ticas**
- **F sica**
- **Qu mica**
- **Historia**
- **Filosof a**

3. Herencia

El proyecto contendr  herencia, pues toda publicaci n puede ser una revista, un peri dico, una novela o un libro de texto. Podr an hacerse m s clases, pero lo dejaremos para subir nota.

Objeto/Clase Publicaci n:

- **Identificador  nico**
 - **T tulo**
 - **N mero p ginas**
 - **Fecha de ingreso en nuestra biblioteca.**
 - **Estado (prestado o no)**
 - **Fecha de devoluci n**
 - **Fecha de publicaci n**
- Contendr  un contador est tico para la generaci n de los id  nicos.

Objeto/Clase novela:

- **Autor**
- **Editorial**
- **GeneroNovela (Enum Temática)**
- **Edicion**

Objeto/Clase periódico:

- **Genero del Periódico (Enum GeneroPeriodico)**
- **Periodo Publicación (Enum Periodo)**

Objeto/Clase libro de texto:

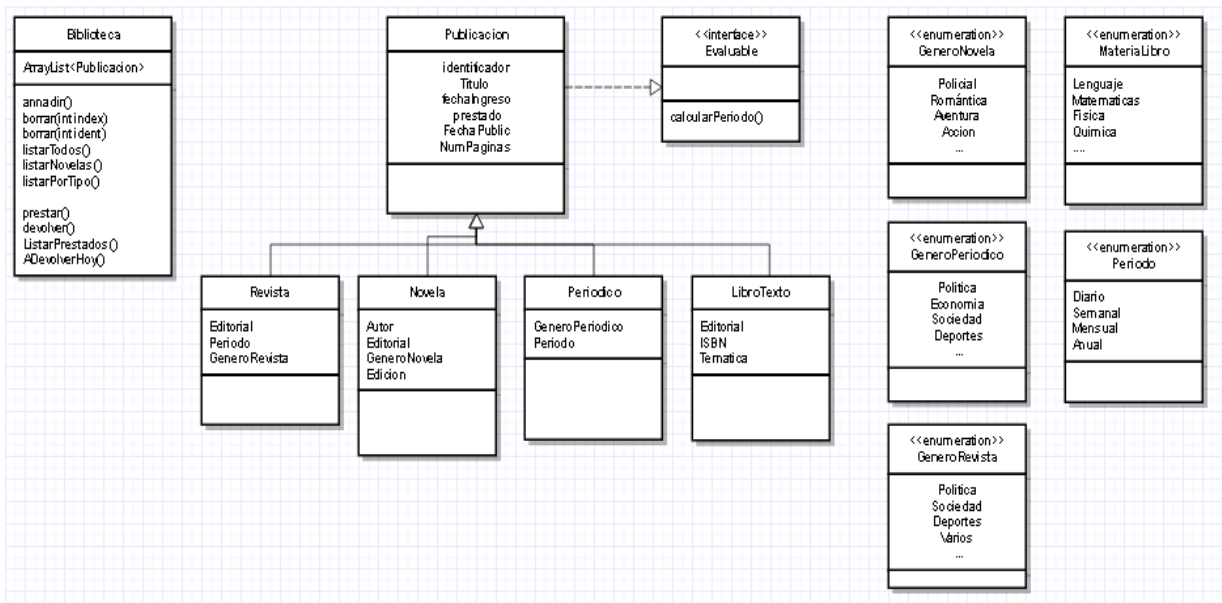
- **Editorial**
- **ISBN**
- **Materia a la que pertenece (Enum Materia)**
- **Edicion**

Objeto/Clase revista:

- **Editorial**
- **Periodo de publicación**
- **Género de la revista (Enum GeneroRevista)**

Esquema general:

Gliffy / *Esquema Biblioteca, v3



4. Interfaces

Serializable y comparable obligatoriamente

- Interfaz Evaluable:

Todas las publicaciones serán evaluables, es decir, cada publicación tendrá una puntuación, la cual se determina por diferentes factores según cada tipo de publicación, por ejemplo:

Se determinará el tiempo de préstamo en función de:

- Novela: número de páginas y fecha de ingreso en la biblioteca.
- Revista: en función de la fecha de publicación y número de páginas
- Periódico: periodo de publicación y fecha de publicación
- Libro de texto: en función del número de páginas solamente.

4.1 Tiempos de préstamo

Tiempo de préstamo para novela:

- Por cada página, se multiplicara por 0,1 y se sumara a una puntuación, y a esto se le sumará 0,5 por cada año que lleve en nuestra biblioteca, de forma que:

- Una novela de 200 páginas que lleva 3 años en nuestra biblioteca:

Puntuación: $(200 \text{ (paginas)} \times 0,1) + (3 \times 0,5) = 20 + 1,5 = \mathbf{21,5 \text{ puntos.}}$

Tiempo de préstamo para una revista:

- Por cada año que haya pasado desde que se publicó, se multiplicará por 0,5 y por cada página, se multiplicará por 0,7. La suma de ambos determinará la puntuación.

- Una revista que lleva 5 años publicada y tiene 50 páginas:

Puntuación: $(5 \text{ años} \times 0,5) + (50 \text{ paginas} \times 0,7) = 2,5 + 35 = \mathbf{37,5 \text{ puntos.}}$

Tiempo de préstamo para un periódico:

- Los periódicos diarios sumaran una puntuación de 30 más 0,5 por cada año transcurrido de su publicación.
- Los periódicos semanales sumaran una puntuación de 40 más 0,5 por cada año transcurrido.
- Los mensuales sumaran 50 puntos más 0,5 por cada año transcurrido
- Y los anuales sumaran 60 puntos más 0,5 por cada año transcurrido.

- Por ejemplo, un periódico que tiene más de 30 años y es diario:

Puntuación: $(30 \text{ años} \times 0,5) + (30 \text{ diario}) = 15 + 20 = \mathbf{45 \text{ puntos.}}$

Tiempo de préstamo para un libro de texto:

- La puntuación se calculará multiplicando cada página por 0,3, por ejemplo:

- Un libro de texto de 200 páginas:

Puntuación: $200 \times 0,3 = \mathbf{60 \text{ puntos}}$

Calculo de los días de préstamo en función de la puntuación:

- Puntuación de 0 a 20: 30 días
- Puntuación de 20 a 40: 15 días
- Puntuación de 40 a 60: 7 días
- Puntuación de 60 a 100: 1 día
- Puntuación de 100 hasta infinito: 12 horas

5. Flujos de datos

Nuestra biblioteca se podrá guardar y recuperar con opciones como:

- **Nuevo**
- **Abrir**
- **Guardar**
- **Guardar como...**

6. Ficheros

Guardaremos la biblioteca completa como un objeto serializable.

Nuestra biblioteca se podrá guardar y recuperar en un archivo con extensión .obj.

7. Excepciones

- CampoVacioException → Para cuando un campo esté vacío
- LibroNoExisteException → Para cuando un libro no exista.
- FechaNoValidaException → Para el control del formato de una fecha
- ISBNIncorrectoException → Para el formato del ISBN de los libros de texto.
- NullPointerException → Habrá otras excepciones que deberemos controlar

Algunas otras excepciones irán surgiendo a medida que vayamos realizando la programación.

8. Expresiones regulares

- Expresión regular para comprobar que los títulos de los libros no están en blanco:
 - Comprueba que el campo contiene un mínimo de dos letras y no está vacío ni contiene espacios al principio ni final.
- Expresión regular para el control del ISBN de los libros de texto.
 - Un ISBN compuesto por 10 dígitos de longitud

- Un ISBN también puede contener 13 dígitos, donde los tres primeros pueden ser 978 o 979. Por ejemplo:

- **ISBN-10: 8481812277**
- **ISBN-13: 9788481812275**

- Expresión regular para la fecha:
 - La fecha de ingreso, así como la fecha de la publicación, tendrá el formato **dd/mm/aaaa**, es decir, dos dígitos para los días, otros dos para el mes y cuatro para el año.

9. Fechas

Tendremos un uso y control de fechas para nuestras publicaciones:

- **Ingreso de un nuevo libro**
- **Realización de un préstamo**
- **Realización de una devolución.**
- **Calculo de periodo de préstamo** en función de su puntuación (Interfaz Evaluable).

10. GUI

Contendrá su entorno gráfico, con una ventana principal donde habrá un menú que nos permitirá las siguientes funciones:

- Archivo:
 - **Nuevo**
 - **Abrir**
 - **Guardar**
 - **Guardar como...**
 - **Salir (controlando si está guardada la biblioteca actual).**
- Edición
 - **Añadir libro**
 - **Borrar libro** → Por título o por identificador

- **Buscar libro** → Por título o por identificador

- Listar
 - **Listar todo**
 - **Listar novelas**
 - **Listar periódicos**
 - **Listar libros de texto**
 - **Listar revistas**

- Prestamos / Devoluciones
 - **Prestar libro**
 - **Devolver libro**

El entorno gráfico podrá gestionarse mediante mnemonics y aceleradores