

Reto - Biblioteca La Pingüinera

La Pingüinera es una biblioteca que necesita un sistema para gestionar los libros y novelas que se encuentran disponibles. Es necesario poder agregar, actualizar, eliminar y obtener los libros y/o novelas en el sistema. También se debe tener un control de acceso para que solo puedan acceder al sistema los asistentes de la biblioteca, pero teniendo en cuenta que siempre habrá un usuario principal, el cual podrá acceder a todo. Los datos de este usuario son:

Nombre: John Doe

Correo: administrador@pingu.com.co

Contraseña: contraseñasegura123456

Rol: ADMINISTRADOR

El dueño de la biblioteca tendrá la posibilidad de agregar a los asistentes para que puedan hacerse cargo del inventario de libros y de los préstamos. Cabe resaltar que los asistentes de la biblioteca no pueden acceder a la información de los usuarios del sistema, solo tendrán acceso al apartado de libros y préstamos. Los asistentes tendrán los mismos campos del dueño de la biblioteca, nombre, correo, contraseña y rol (ASISTENTE). Los libros van a tener la siguiente información: título, autor, área del conocimiento, número de páginas, cantidad de ejemplares, cantidad prestados y la cantidad disponible. Mientras que las novelas van a tener la siguiente información: título, autor, género, edad de lectura sugerida, cantidad de ejemplares, cantidad de prestados y cantidad disponible.

El sistema debe brindarnos la posibilidad de listar a todos los autores que se encuentran asociados a los libros/novelas. Esto con el fin de filtrar todos los libros o todas las novelas que son de un autor en específico. Como punto a tener en cuenta, las novelas y los libros deben gestionarse por separado, ya que son elementos diferentes dentro de La Pingüinera. Los préstamos dentro de la biblioteca deben realizarse de la siguiente manera: Para que un usuario pueda prestar un libro, tendrá que crear una cuenta en el sistema de la biblioteca o, en caso de ya tener una cuenta, deberá iniciar sesión con su correo y contraseña. El sistema debe identificar a todos los prestamistas como "LECTORES" y los ejemplares que se muestran a este grupo de usuarios debe ser exclusivamente la lista de libros/novelas cuya cantidad disponible sea mayor que 0. Posteriormente, tendrá que seleccionar los libros y novelas que quiera prestar. El sistema debe pedirle la confirmación al usuario para finalizar el proceso. Si el usuario confirma el préstamo, debe descontar 1 a la cantidad de unidades disponibles en todos los libros y novelas que se escogieron. Se debe recibir la fecha del día del préstamo y cuándo se regresarán los ejemplares (este intervalo de tiempo puede ser máximo de 15 días). En el momento en que se confirma el préstamo, el usuario debe dirigirse hacia donde uno de los asistentes de la biblioteca para que le hagan entrega de los ejemplares, por lo tanto, el sistema debe permitirle al asistente consultar los préstamos que ha realizado el usuario cuando lo consulta por medio de su correo. Esto con el fin de poder confirmar el préstamo con algún funcionario de la biblioteca y que se puedan entregar los ejemplares.

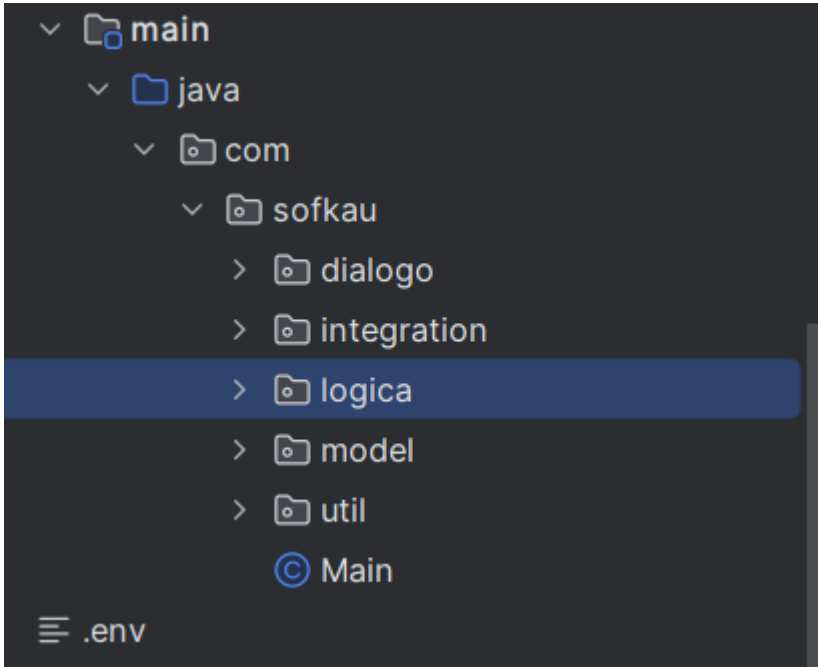
Los préstamos deben tener 3 estados:

Solicitado (Se debe asignar este estado cuando el usuario realiza el préstamo a través de la plataforma). Realizado (Se debe asignar este estado cuando el asistente confirma el préstamo y entrega los ejemplares). Finalizado (Se debe asignar este estado cuando el usuario devuelva los ejemplares). Una vez el usuario esté haciendo la devolución en la biblioteca, el sistema debe permitirle al asistente buscarlo por su correo y, una vez confirmada la devolución de todos los ejemplares, debe aumentar 1 a la cantidad disponible de cada uno, actualizarlo en el listado de libros y/o novelas y marcar el préstamo como finalizado. Adicionalmente, si la transacción se realiza después de la fecha de devolución establecida, el sistema debe arrojar una advertencia de que el usuario no cumplió con la fecha establecida.

Aspectos técnicos a tener en cuenta

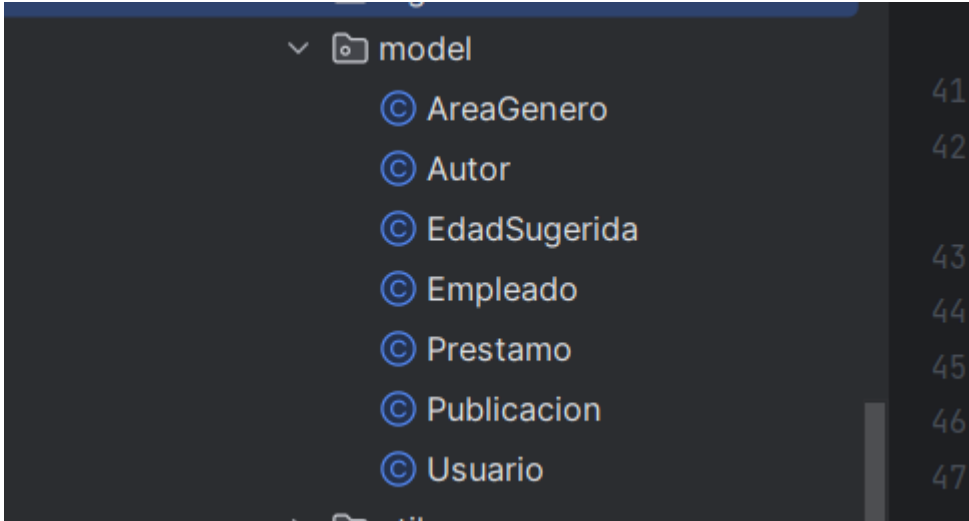
Este reto debe ser realizado en una aplicación de consola, NO SE DEBE REALIZAR FRONTEND, pero sí debe realizar la persistencia de la información en base de datos. Los temas a potenciar acá son la programación orientada a objetos y el manejo de colecciones. Lo que se va a realizar con los libros/novelas al tener cantidad prestada, cantidad de ejemplares y cantidad disponible es conocido como un borrado lógico. Se le conoce así porque el registro no va a ser eliminado del sistema en ningún momento, pero para el usuario no va a ser visible porque no cuenta con unidades disponibles para poder prestar, y por lo mismo, no le vamos a mostrar un ejemplar que no pueda prestar.

Disposición carpetas



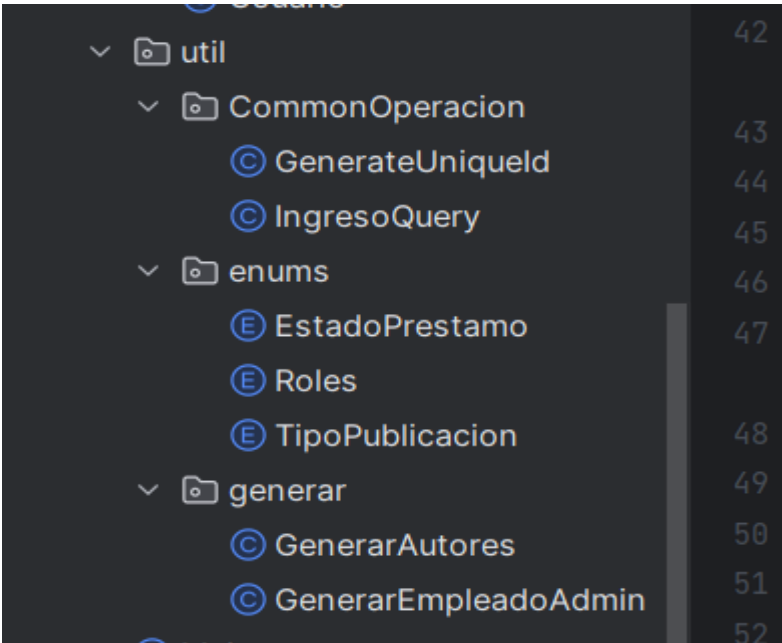
Model

Esta carpeta contiene todos los modelos utilizados para la solución.



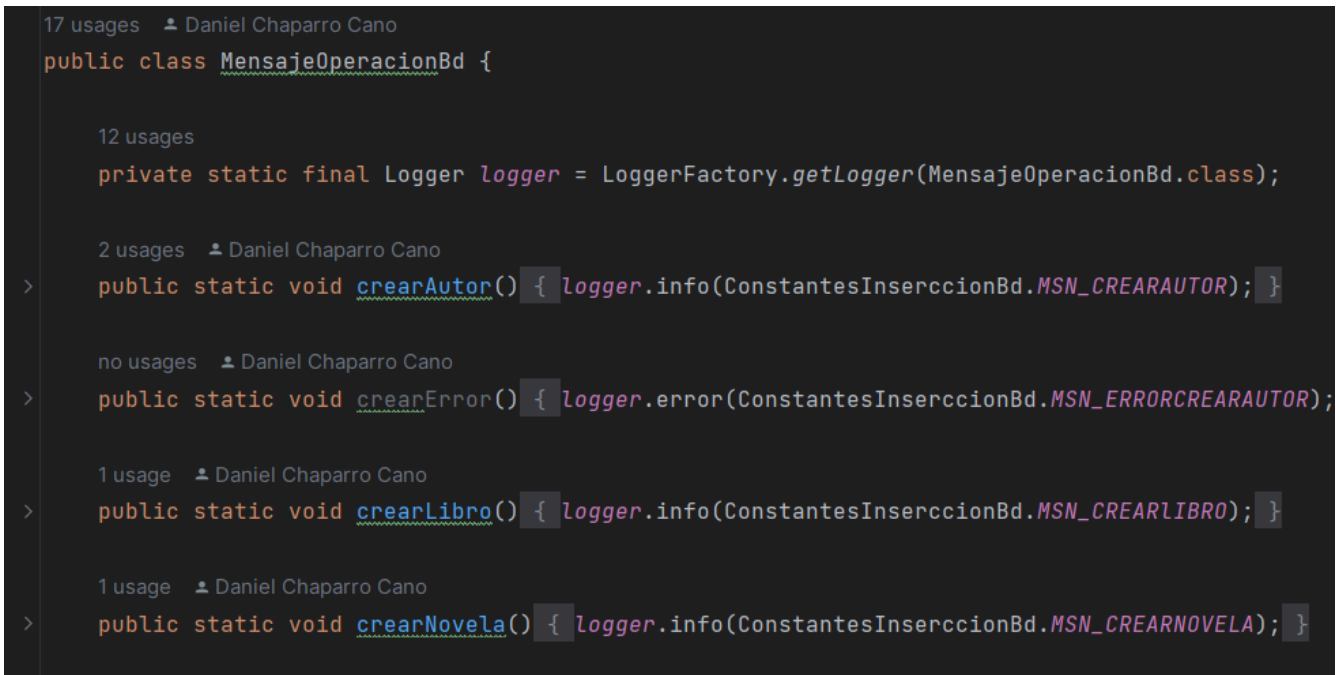
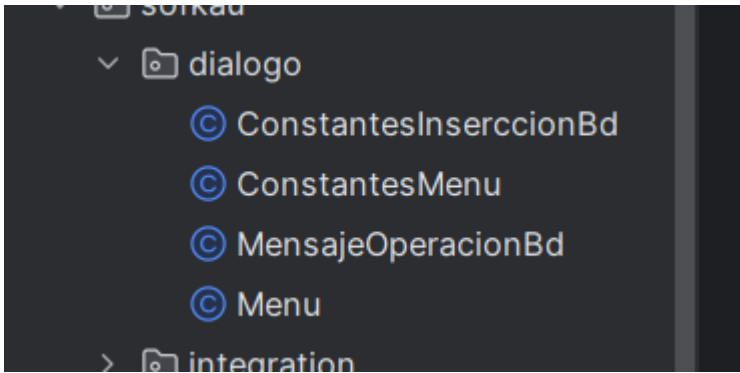
Util

Esta carpeta tiene 3 sub carpetas que contienen clases generales y enums, de uso común, en todo el proyecto.



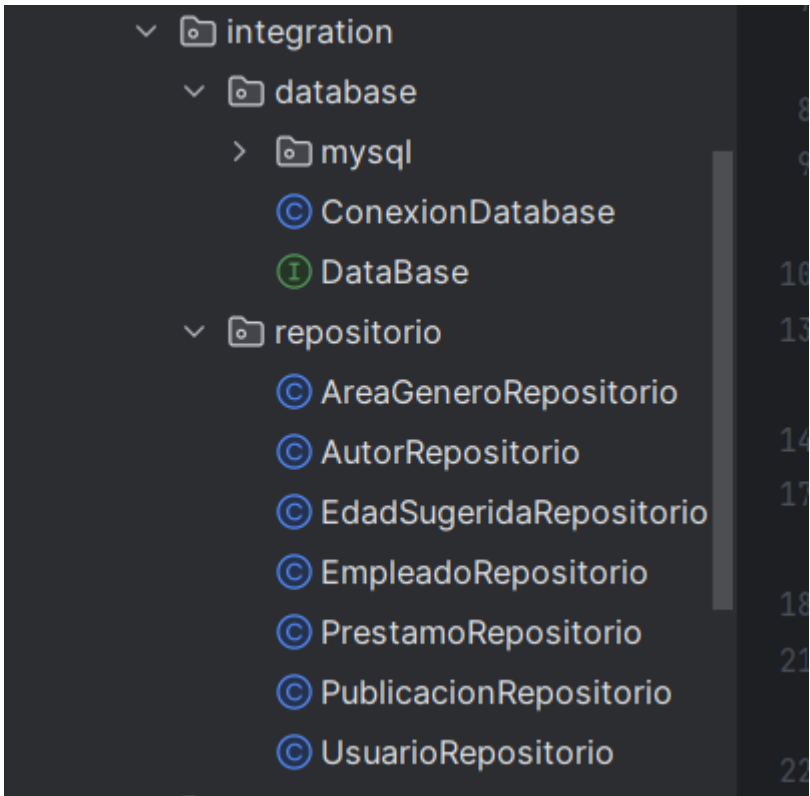
Diálogo

Esta carpeta posee métodos reutilizables para imprimir por pantalla según sea el caso, además de Logger para las alertas de las inserciones en la base de datos.



Integracion

Esta carpeta posee dos sub carpetas ambas relacionadas a la integración con la base de datos,



La clase ConexionDatabase Maneja la conexión a la base de datos, así mismo devuelve un objeto para la inserccion.

La carpeta repositorio es una capa entre la base de datos y la lógica manejando la interacción entre ambos.

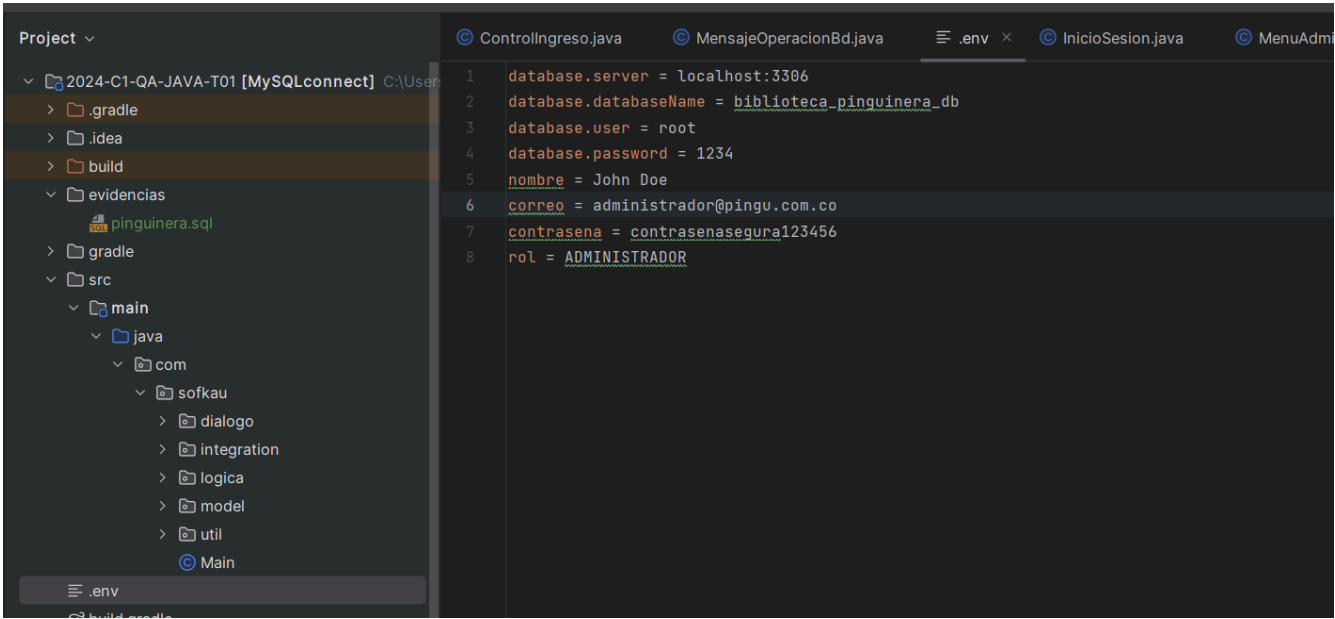
Lógica

Esta carpeta tiene la lógica del aplicativo, gestionando el ingreso de los datos y las operaciones e interacción entre las distintas clases.

logica	
Autor	8
control	9
ControllIngreso	10
InicioSesion	13
MenuAdministrador	
MenuAsistente	14
MenuEmpleado	17
MenuPropietario	
MenuUsuario	18
empleado	21
prestamo	
publicacion	22
usuario	25

logica		
Autor	6	17 usages - Daniel O
AutorOperaciones	7	public class Mens
control		12 usages
ControllIngreso	8	private stati
InicioSesion	9	
MenuAdministrador	10	2 usages - Dani
MenuAsistente	13	public static
MenuEmpleado		
MenuPropietario	14	no usages - Dar
MenuUsuario	17	public static
empleado		1 usage - Daniel
EmpleadoOperaciones	18	public static
prestamo	21	
PrestamoOperaciones		1 usage - Daniel
publicacion	22	public static
AreaGeneroOperaciones	25	
EdadSugeridaOperaciones	26	1 usage - Daniel
PublicacionOperaciones	29	public static
usuario		1 usage - Daniel
UsuarioOperaciones	30	public static
model	33	

.Env



Por último se tiene el .env para manejar credenciales y variables locales del aplicativo.