

Caso de Estudio: Una biblioteca universitaria quiere digitalizar su sistema de préstamo de libros. Para ello, se necesita modelar una solución que permita gestionar libros, usuarios, préstamos y devoluciones.

Requisitos del sistema:

1. La biblioteca tiene usuarios que pueden ser estudiantes o profesores.
2. Cada usuario tiene un ID, nombre, correo y tipo (estudiante/profesor).
3. Los libros tienen un ID, título, autor y categoría.
4. Un usuario puede tomar prestado varios libros, pero un libro solo puede estar prestado a un usuario a la vez.
5. Un préstamo tiene una fecha de inicio y una fecha de devolución estimada.
6. Cuando un usuario devuelve un libro, se registra la fecha real de devolución.
7. Un profesor puede reservar libros con anticipación, pero los estudiantes no.
8. Se necesita registrar un historial de préstamos por usuario.

Stakeholders:

Internos:

Estudiantes

Profesores

Director de la biblioteca universitaria

Rector de la universidad

Analista

Externos:

Bibliotecarios de la universidad

Visitantes de la biblioteca

Requisitos Funcionales:

Código	Requerimiento Funcional
RQF001	Nombre: Preferencia de reserva de libros para profesores
	Descripción: Habilitar la reserva anticipada de libros para los profesores.

	Usuarios: profesores.
RQF002	Nombre: historial de prestamos
	Descripción: Registrar los prestamos realizados por un usuario a lo largo del tiempo
	Usuarios: profesores, estudiantes.
RQF003	Nombre: Reserva única de libro
	Descripción: Deshabilitar el préstamo de un libro si este ya está bajo préstamo de un usuario
	Usuarios: estudiantes, profesores.

HU001	Nombre: Preferencia de reserva de libros para profesores
	Historia: La biblioteca universitaria debe permitir a sus profesores reservar libros para uso como material en sus clases, debido a esto, los profesores que planeen usar un libro en un futuro deben poder reservar su libro con antelación.
	Criterios de aceptación: <ol style="list-style-type: none"> 1. Permitir reservar los libros con antelación. 2. Priorizar las reservas hechas con antelación. 3. Limitar las reservas con antelación para profesores.

Módulos:

Usuario

Estudiante: Se hereda de usuario

Profesor: Se hereda de usuario, puede reservar con antelación

Reservar: Permite realizar una reserva con antelación

Prestamo: La reserva, registra los datos del prestamo, incluyendo fecha inicial y final del préstamo

Libro

Devolución: Datos relacionados a las devoluciones realizadas

Interpretación diagrama de clases

El diagrama cuenta con 7 clases

Usuario:

Atributos:

- idUsuario

- nombreUsuario
- correoUusario
- tipoUsuario

atributo clave: idUsuario

Operaciones:

- registrarUsuario()
- consultarUsuario()
- desactivarUsuario()

Relaciones:

Herencia con Estudiante y Profesor

Estudiante:

Atributos: Todos sus atributos son heredados de Usuario

atributo clave: idUsuario (heredado de la clase usuario)

Operaciones:

Todas sus operaciones son heredadas de Usuario

Profesor:

Atributos: Todos sus atributos son heredados de Usuario

atributo clave: idUsuario (heredado de la clase usuario)

Operaciones:

Todas sus operaciones son heredadas de Usuario

Reservar:

Atributos:

- idReserva

atributo clave: idReserva

Operaciones:

realizarReserva()

Prestamos:

Atributos:

- idPrestamo
- fechaInicio
- fechaFinal

atributo clave: idPrestamo

Operaciones:

- registrarPrestamo()
- consultarPrestamo()
- desactivarPrestamo()
- reportarRetraso()

Libro:

Atributos:

- idLibro
- tituloLibro
- autorLibro
- categoriaLibro

atributo clave: idLibro

Operaciones:

- registrarLibro()
- consultarLibro()
- desactivarLibro()

Devolución:

Atributos:

- idDevolución
- fechaDevolucion

atributo clave: idDevolucion

Operaciones:

registrarDevolucion()

consultarDevolucion()

Diagrama de Clases

