



Análisis y Diseño de Sistemas de Información
Ingeniería Informática de Gestión y Sistemas de
Información

Biblioteca

Autores:

Xabier Gabiña Barañano
Janire Veganzones García
Ainhize Martinez Duran
Javier Criado Garcia
Kepa Reches Urrutia
Jon Valdes Granado
Mohamed El Basri Dehbi

27 de noviembre de 2023

Índice general

Changelog

1.1. Versión 1 (2023-10-08)

- Añadido caso de uso general.
- Añadidos los casos de uso extendidos.

1.2. Versión 2 (2023-10-22)

- Añadido modelo de dominio.
- Modificado el caso de uso general.
- Modificados los casos de uso extendidos.

1.3. Versión 3 (2023-11-05)

- Añadido plan de Pruebas.
- Modificado el modelo de dominio.

1.4. Versión 4 (2023-11-15)

- Añadido diagrama relacional.
- Añadido diagrama de clases.
- Añadido diagrama de comunicación.

Introducción

En un mundo cada vez más digitalizado, las bibliotecas modernas están buscando formas innovadoras de adaptarse a las necesidades cambiantes de sus usuarios. La biblioteca que nos ocupa se encuentra en este proceso de transformación, reconociendo la importancia de expandir sus servicios en línea y brindar a sus usuarios una experiencia más enriquecedora y conectada. Para llevar a cabo esta ambiciosa misión, han solicitado la colaboración de nuestro equipo de desarrollo.

En su estado actual, la biblioteca cuenta con un servicio web que permite a los usuarios consultar su extenso catálogo de libros. Además, ofrece un apartado de usuarios que pueden acceder a una sección privada. Sin embargo, esta plataforma carece de funcionalidades adicionales que pueden mejorar significativamente la experiencia de los usuarios y promover una mayor interacción entre ellos.

En este proyecto, nuestro objetivo es expandir y mejorar el servicio web de la biblioteca, dotándolo de una serie de funcionalidades clave. Estas nuevas características incluyen la gestión de reservas, la capacidad de dejar reseñas y comentarios en los libros, la creación de una red de amigos entre los usuarios, la introducción de un rol de administrador con capacidades de gestión, la implementación de foros para discusión y la incorporación de sistemas de recomendación, tanto basados en el historial de lecturas como en la afinidad de intereses entre usuarios.

A lo largo de este documento, exploraremos en detalle cada una de estas funcionalidades, presentando soluciones técnicas y estratégicas para su implementación. Este proyecto representa una emocionante oportunidad para ayudar a la biblioteca a evolucionar y brindar un servicio en línea más completo y enriquecedor a su comunidad de usuarios ávidos de conocimiento y lectura.

Casos de Uso

3.1. General

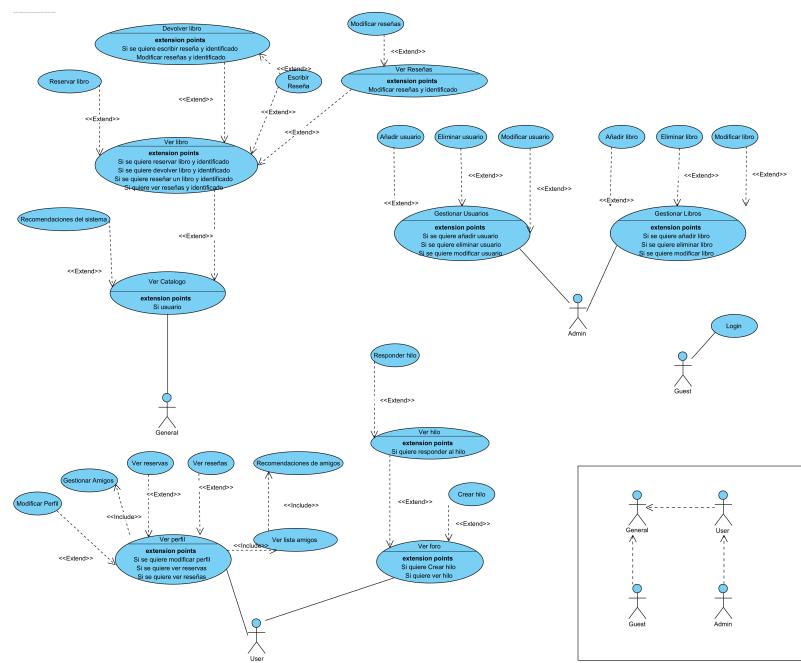


Figura 3.1: Diagrama de casos de uso general

3.2. Extendidos

3.2.1. Gestión de reservas

Responsable: Mohamed El Basri
<pre>graph LR; General --- VerCatalogo[Ver catálogo extension points Si usuario]; VerCatalogo --> VerLibro[Ver libro extension points Si se quiere reservar libro e identificado]; VerLibro --> ReservarLibro[Reservar libro]</pre>
Figura 3.2.1.1: Diagrama de casos de uso de la subfuncionalidad de gestión de reservas
Nombre: Ver libros recomendados
Descripción: Permite a los usuarios realizar reservas de libros.
Actores: Usuario
Precondiciones: Estar identificado en el sistema.
Requisitos no funcionales: Ninguno
Flujo de Eventos: <ol style="list-style-type: none">El usuario selecciona el libro deseado en el catálogo y pulsa “Reservar”. Se abre la interfaz de reserva (Figura 3.2.1.2).Si el usuario pulsa aceptar, si el tiempo de reserva es menor a 2 meses aparece una interfaz indicando que la reserva se ha realizado con éxito (Figura 3.2.1.3).<ol style="list-style-type: none">Si el usuario pulsa aceptar, si el tiempo de reserva es mayor a 2 meses aparece una interfaz indicando error (Figura 3.2.1.4).Si el usuario pulsa cancelar, se cierra la interfaz.
Poscondiciones: Ninguna
Interfaz Gráfica:



Figura 3.2.1.2: Interfaz de reserva



Figura 3.2.1.3: Interfaz de reserva con éxito

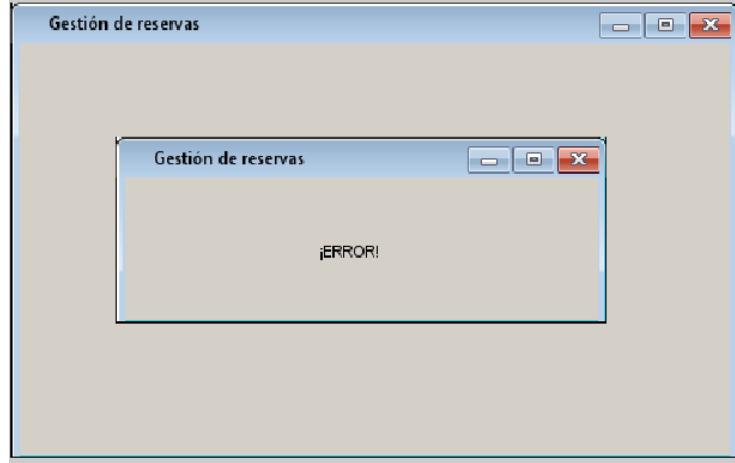


Figura 3.2.1.4: Interfaz de reserva con error

3.2.2. Reseñas

Responsable: Javier Criado
<pre> graph LR General --> VerCatalogo([Ver Catalogo]) VerCatalogo -- "extension point Si usuario" --> VerLibro([Ver libro]) VerLibro -- "extension point Si se quiere devolver libro y identificado" --> DevolverLibro([Devolver libro]) VerLibro -- "extension point Si se quiere escribir reseña y identificado" --> EscribirReseña([Escribir Reseña]) VerLibro -- "extension point Si se quiere ver reseñas y identificado" --> ModificarReseña([Modificar reseñas]) </pre>
<p>Nombre: Reseñas</p> <p>Descripción: Permite al cliente puntuar y comentar en un libro una vez lo haya devuelto, también permite modificar estos datos posteriormente.</p>
<p>Actores: Usuario</p> <p>Precondiciones: Estar identificado en el sistema</p> <p>Requisitos no funcionales: Ninguno</p>
<p>Flujo de Eventos:</p> <ol style="list-style-type: none"> 1. El usuario entra en el catalogo para ver los libros disponibles. <ol style="list-style-type: none"> a) El usuario selecciona uno de los libros. [Figura 3.2.2.2] <ol style="list-style-type: none"> 1) Al seleccionar 'Devolver libro', se devolvera el libro y se podra añadir una nueva reseña. 2) Al seleccionar 'Escribir reseña', se abrirá una pestaña donde se podrá escribir una reseña sobre el libro. [Figura 3.2.2.3] 3) Al seleccionar el boton 'Ver reseñas', el usuario podrá ver las reseñas sobre el libro. [Figura 3.2.2.4] <ol style="list-style-type: none"> a' Al hacer click sobre una de las reseñas esta se mostrara en una ventana aparte. [Figura 3.2.2.5]
<p>Poscondiciones: Ninguna</p> <p>Interfaz Gráfica:</p>

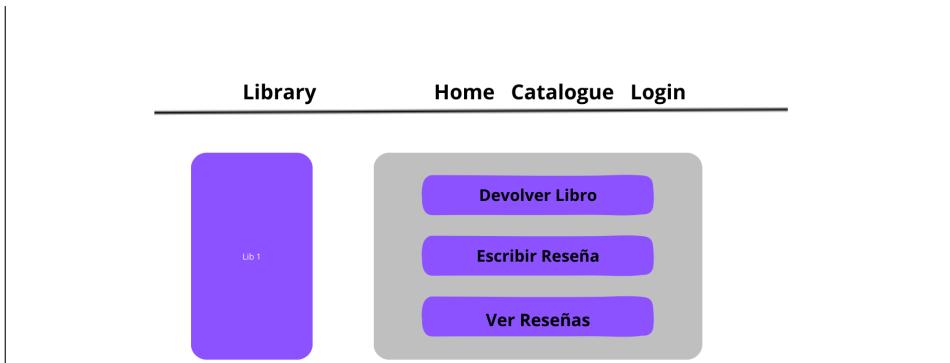


Figura 3.2.2.2: Interfaz de ver libro.

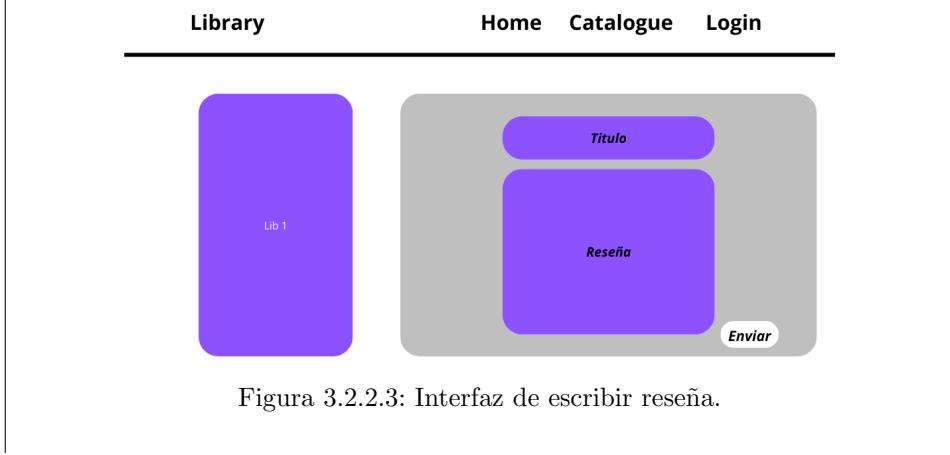


Figura 3.2.2.3: Interfaz de escribir reseña.

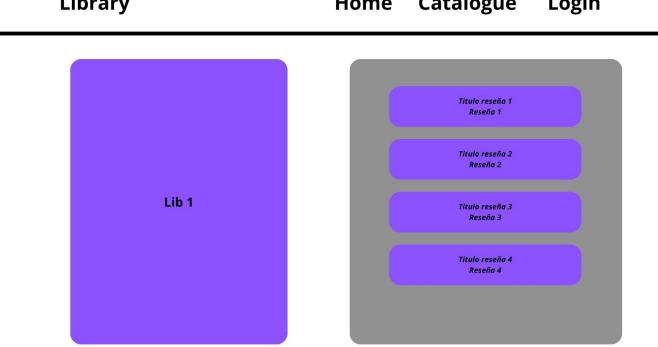


Figura 3.2.2.4: Interfaz para ver las reseñas.

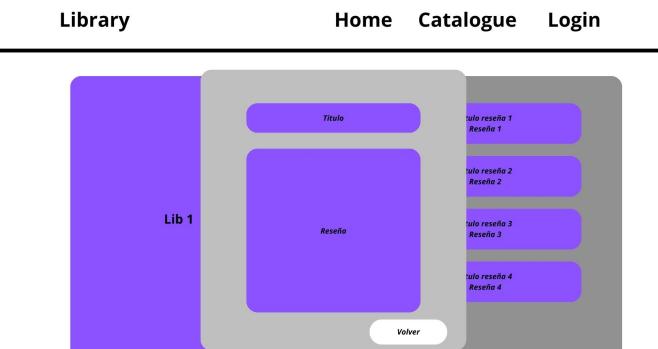


Figura 3.2.2.5: Interfaz para ver una reseña.

3.2.3. Red de amigos

Responsable: Jon Valdes
<pre> classDiagram actor User useCase VerPerfil useCase ModificarPerfil useCase GestionarAmigos useCase VerReservas useCase VerReseñas useCase VerListaAmigos User --> VerPerfil VerPerfil <<Extend>> ModificarPerfil VerPerfil <<Extend>> GestionarAmigos VerPerfil <<Extend>> VerReservas VerPerfil <<Extend>> VerReseñas VerPerfil <<Include>> VerListaAmigos VerReservas <<Extend>> VerReseñas </pre>
Figura 3.2.3.1: Diagrama de casos de uso de la subfuncionalidad de red de amigos
Nombre: Red de amigos.
Descripción: Permite al usuario gestionar su perfil, de modo que puede tanto consultar el historial de reservas y de reseñas, como añadir o eliminar amigos. También puede editar sus datos personales.
Actores: Usuario.
Precondiciones: Estar identificado en el sistema.
Requisitos no funcionales: Ninguno
Flujo de Eventos: <ol style="list-style-type: none"> 1. El usuario accede a su perfil. [Figura 3.2.3.2] <ol style="list-style-type: none"> a) El usuario puede pulsar 'Editar perfil' para editar sus datos. [Figura 3.2.3.5] b) El usuario puede pulsar 'Consultar reservas' para ver su historial de reservas. c) El usuario puede pulsar 'Consultar reseñas' para ver su historial de reseñas. d) El usuario puede ver su lista de amigos. e) El usuario puede pulsar 'Añadir amigo' para añadir un amigo. [Figura 3.2.3.3] f) El usuario puede pulsar 'Solicitudes de amistad' para ver las solicitudes de amistad pendientes. g) El usuario puede pulsar 'Eliminar amigo' para eliminar un amigo. [Figura 3.2.3.4]
Poscondiciones: Ninguna
Interfaz Gráfica:

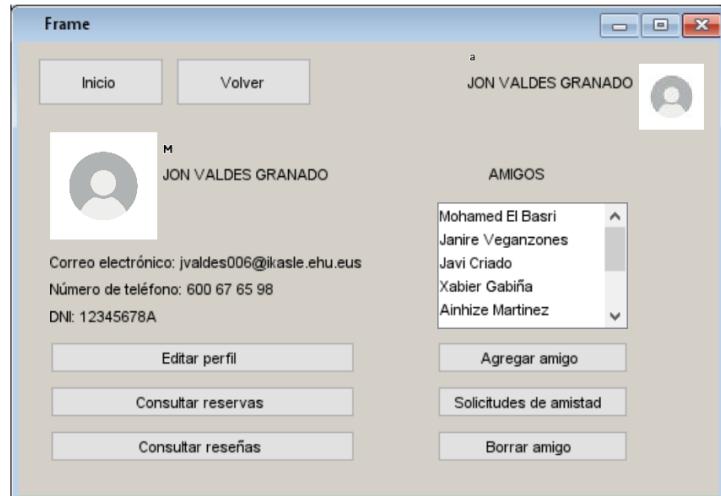


Figura 3.2.3.2: Interfaz del perfil de usuario.

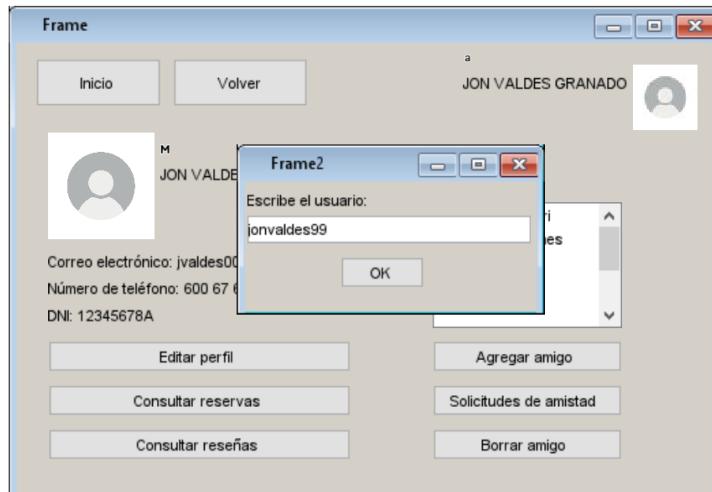


Figura 3.2.3.3: Interfaz para añadir un amigo.

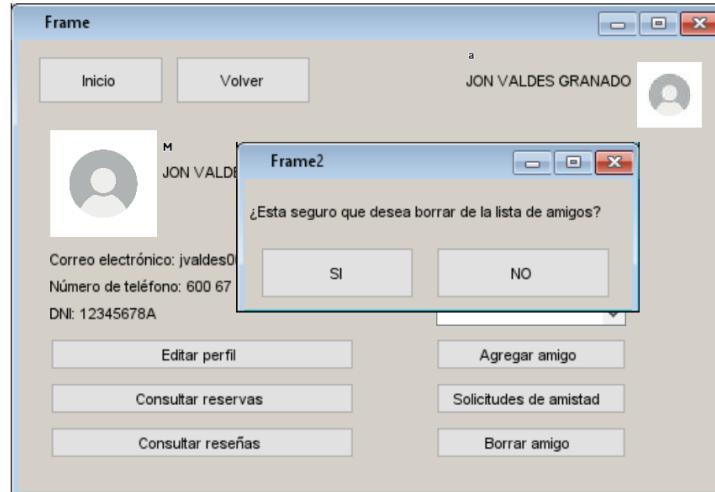


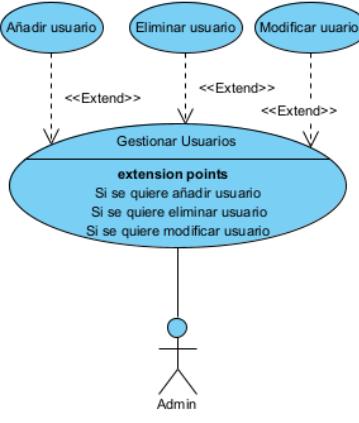
Figura 3.2.3.4: Interfaz de alerta para borrar un amigo.



Figura 3.2.3.5: Interfaz para editar el perfil.

3.2.4. Administrador

Gestión de usuarios

Responsable: Janire Veganzones

Figura 3.2.4.1.1: Diagrama de casos de uso de la subfuncionalidad de gestión de usuarios
Nombre: Gestionar usuarios
Descripción: Permite crear, eliminar o modificar un usuario en el sistema.
Actores: Admin
Precondiciones: Estar identificado en el sistema y ser administrador.
Requisitos no funcionales: Ninguno

Flujo de Eventos:

1. Se accede al menu de administrador. [Figura 3.2.4.1.2]
 - a) Se selecciona la opción de gestionar usuarios. [Figura 3.2.4.1.3]
 - 1) El administrador pulsa 'Crear' para crear un nuevo usuario. [Figura 3.2.4.1.4]
 - a' Si hay un error al crear un usuario saldra un error mostrandolo. [Figura 3.2.4.1.5]
 - 2) El administrador pulsa 'Borrar' para borrar un usuario. [Figura 3.2.4.1.6]
 - a' Si el usuario a borrar no existe mostrara un error. [Figura 3.2.4.1.7]
 - 3) El administrador pulsa 'Modificar' para modificar un usuario. [Figura 3.2.4.1.8]
 - a' Si hay un error al modificar un usuario saldra un error mostrandolo. [Figura 3.2.4.1.9]

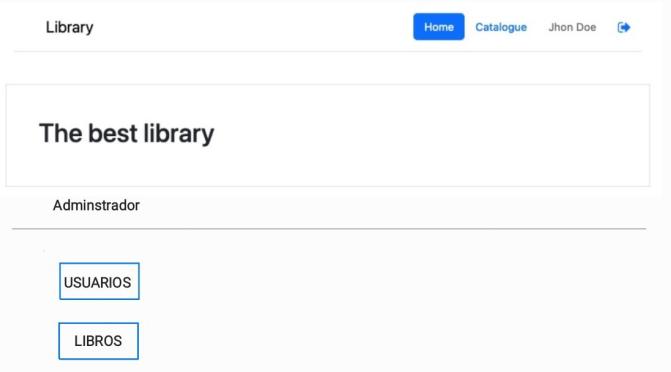
Poscondiciones: Ninguna**Interfaz Gráfica:**

Figura 3.2.4.1.2: Interfaz del menú de administrador.

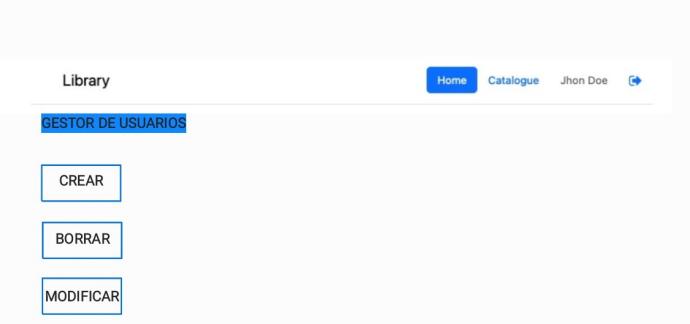


Figura 3.2.4.1.3: Interfaz del menu de gestión de usuarios.

The screenshot shows a 'CREAR USUARIO' form. The form fields are labeled 'Nombre', 'Apellidos', 'Fecha nacimiento', 'Email', 'Contraseña', and 'Repetir contraseña'. Each label is followed by a text input field. Below the input fields is a blue rectangular button labeled 'CREAR'.

Figura 3.2.4.1.4: Interfaz para añadir un usuario.

The screenshot shows a web application interface for creating a new user. At the top, there is a navigation bar with links for 'Home', 'Catalogue', and 'Login'. Below this, the main content area has a title 'CREAR USUARIO' in bold. The form consists of six input fields: 'Nombre' (Name), 'Apellidos' (Last Name), 'Fecha nacimiento' (Date of Birth), 'Email', 'Contraseña' (Password), and 'Repetir contraseña' (Repeat Password). A message box at the bottom left states 'Alguno de los campos introducidos es incorrecto' (One or more of the entered fields are incorrect). To the right of this message is a blue-bordered button labeled 'CREAR'.

Figura 3.2.4.1.5: Interfaz de error al añadir un usuario.

The screenshot shows a web application interface titled "Library". At the top right, there are navigation links: "Home" (highlighted in blue), "Catalogue", "Jhon Doe", and a user icon. Below this, a blue button labeled "BORRAR USUARIO" is visible. A form field labeled "Email" contains the placeholder text "Email". To the right of the email field is a blue "Borrar" button.

Figura 3.2.4.1.6: Interfaz para borrar un usuario.

This screenshot shows the same "Library" interface as the previous one. The "BORRAR USUARIO" button is now highlighted in blue. The "Email" field is empty. Below the field, a red message box displays the text "El usuario introducido no existe" (The user entered does not exist). To the right of the message is a blue "Borrar" button.

Figura 3.2.4.1.7: Interfaz de error al borrar un usuario.

The screenshot shows a web-based application interface for modifying a user. At the top, there is a navigation bar with links for 'Home', 'Catalogue', and 'Login'. Below this, a title bar reads 'MODIFICAR EL USUARIO'. The main area contains a form with the following fields:

- Nombre: A text input field.
- Apellidos: A text input field.
- Fecha nacimiento: A text input field.
- Email: A text input field.
- Nuevo nombre: A text input field.
- Nuevos apellidos: A text input field.
- Nueva fecha nacimiento: A text input field.
- Nueva contraseña: A text input field.
- Repetir nueva contraseña: A text input field.

At the bottom right of the form is a blue rectangular button labeled 'CREAR'.

Figura 3.2.4.1.8: Interfaz para modificar un usuario.

Library

Home Catalogue Login

MODIFICAR EL USUARIO

Nombre

Apellidos

Fecha nacimiento

Email

Nuevo nombre

Nuevos apellidos

Nueva fecha nacimiento

Nueva contraseña

Repetir nueva contraseña

Alguno de los campos introducidos es incorrecto

CREAR

The screenshot shows a user interface for modifying a user. At the top, there are navigation links: 'Library', 'Home' (which is highlighted in blue), 'Catalogue', and 'Login'. Below this is a section titled 'MODIFICAR EL USUARIO'. It contains several input fields for personal information: 'Nombre', 'Apellidos', 'Fecha nacimiento', 'Email', 'Nuevo nombre', 'Nuevos apellidos', 'Nueva fecha nacimiento', 'Nueva contraseña', and 'Repetir nueva contraseña'. A red rectangular box highlights the message 'Alguno de los campos introducidos es incorrecto' (One or more of the entered fields are incorrect). To the right of this message is a blue rectangular button labeled 'CREAR'.

Figura 3.2.4.1.9: Interfaz de error al modificar un usuario.

Gestión de libros

Responsable: Janire Veganzones
<p>The diagram illustrates the 'Gestionar Libros' extension point. At the top, three use cases are shown: 'Añadir libro', 'Eliminar libro', and 'Modificar libro'. Dashed arrows labeled '<<Extend>>' connect each of these to a central oval labeled 'Gestionar Libros'. This central oval contains the text 'extension points' followed by three items: 'Si se quiere añadir libro', 'Si se quiere eliminar libro', and 'Si se quiere modificar libro'. A vertical line connects the bottom of this oval to a participant symbol labeled 'Admin' at the bottom.</p>
Figura 3.2.4.2.1: Diagrama de casos de uso de la subfuncionalidad de gestión de libros
Nombre: Gestionar libros
Descripción: Permite añadir, eliminar o modificar un libro en el sistema.
Actores: Admin
Precondiciones: Estar identificado en el sistema y ser usuario.
Requisitos no funcionales: Ninguno

Flujo de Eventos:

1. Se accede al menu de administrador. [Figura 3.2.4.2.2]
 - a) Se selecciona la opción de gestionar libros. [Figura 3.2.4.2.3]
 - 1) El administrador pulsa 'Crear' para crear un nuevo libro. [Figura 3.2.4.2.4]
 - a' Si hay un error al crear un libro saldra un error mostrandolo. [Figura 3.2.4.2.5]
 - 2) El administrador pulsa 'Borrar' para borrar un libro. [Figura 3.2.4.2.6]
 - a' Si el libro a borrar no existe mostrara un error. [Figura 3.2.4.2.7]
 - 3) El administrador pulsa 'Modificar' para modificar un libro. [Figura 3.2.4.2.8]
 - a' Si hay un error al modificar un libro saldra un error mostrandolo. [Figura 3.2.4.2.9]

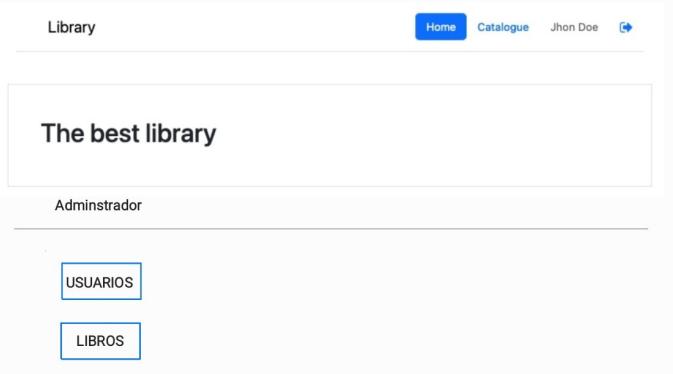
Poscondiciones: Ninguna**Interfaz Gráfica:**

Figura 3.2.4.2.2: Interfaz del menú de administrador.

The screenshot shows a web-based library management system. At the top, there is a navigation bar with links for 'Home', 'Catalogue', 'Jhon Doe', and a user icon. Below the navigation bar, the title 'Library' is displayed. Underneath the title, the heading 'GESTOR DE LIBROS' is shown in blue. Below this heading are three buttons: 'AÑADIR' (Add), 'BORRAR' (Delete), and 'MODIFICAR' (Modify).

Figura 3.2.4.2.3: Interfaz del menu de gestión de libros.

The screenshot shows a 'INSERTAR UN LIBRO' (Insert Book) form. It contains two input fields: 'Título' (Title) and 'Autor' (Author). Below these fields is a large blue 'INSERTAR' (Insert) button.

Figura 3.2.4.2.4: Interfaz para añadir un libro.

The screenshot shows an error message in a red-bordered box stating 'El libro introducido ya existentes' (The book entered already exists). This message is positioned next to the 'INSERTAR' (Insert) button from the previous screenshot.

Figura 3.2.4.2.5: Interfaz de error al añadir un libro.

Library

Home Catalogue Jhon Doe

ELIMINAR UN LIBRO

Título

Autor

ELIMINAR

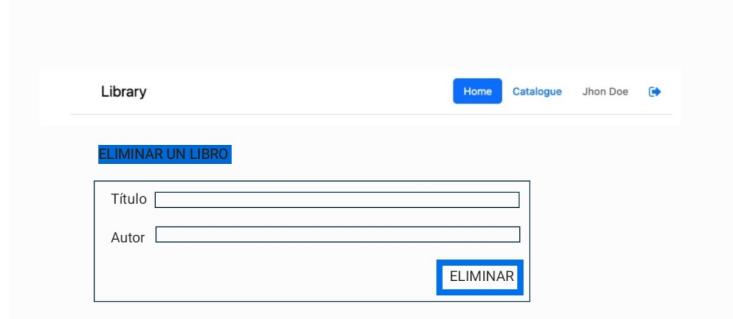


Figura 3.2.4.2.6: Interfaz para borrar un libro.

Library

Home Catalogue Jhon Doe

ELIMINAR UN LIBRO

Título

Autor

El libro introducido no existe

ELIMINAR

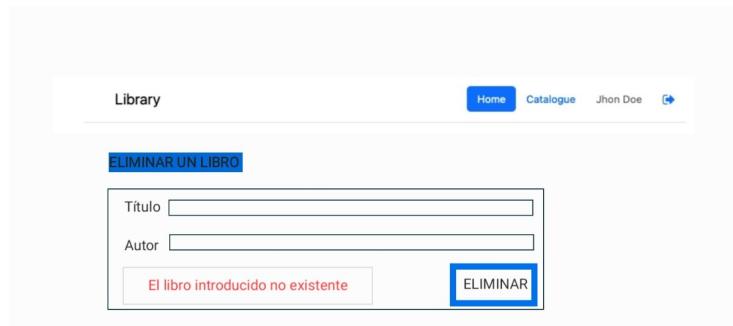


Figura 3.2.4.2.7: Interfaz de error al borrar un libro.

Library

Home Catalogue Jhon Doe

MODIFICAR UN LIBRO

Título anterior

Autor anterior

Título nuevo

Autor nuevo

MODIFICAR

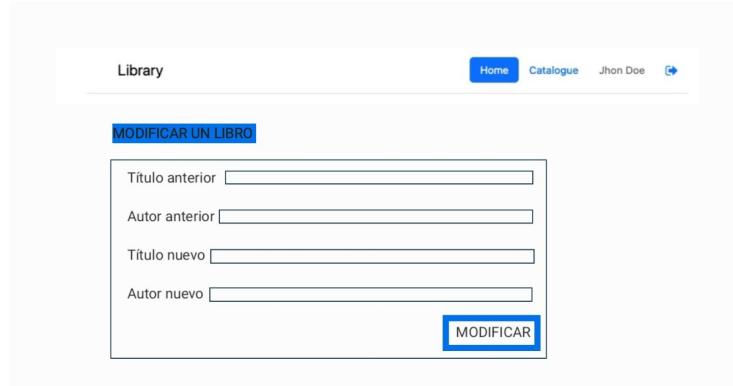


Figura 3.2.4.2.8: Interfaz para modificar un libro.

The screenshot shows a web-based library application interface. At the top, there is a navigation bar with links for 'Home', 'Catalogue', and 'Jhon Doe'. Below the navigation bar, the main content area has a title 'MODIFICAR LIBRO' (Modify Book). There are four input fields: 'Título anterior' (Previous Title), 'Autor anterior' (Previous Author), 'Título nuevo' (New Title), and 'Autor nuevo' (New Author). A red error message box contains the text 'Alguno de los campos introducidos es incorrecto' (One or more of the entered fields are incorrect). To the right of the error message is a blue 'MODIFICAR' (Modify) button.

Library

Home Catalogue Jhon Doe

MODIFICAR LIBRO

Título anterior

Autor anterior

Título nuevo

Autor nuevo

Alguno de los campos introducidos es incorrecto

MODIFICAR

Figura 3.2.4.2.9: Interfaz de error al modificar un libro.

3.2.5. Foros

Responsable: Ainhize Martinez
<pre> classDiagram actor User useCase ResponderHilo useCase VerHilo useCase VerForo useCase CrearHilo User --> ResponderHilo User --> VerHilo User --> VerForo User --> CrearHilo ResponderHilo --> VerHilo : <<Extend>> VerHilo --> VerForo : <<Extend>> VerHilo --> CrearHilo : <<Extend>> class VerHilo { extension points Si quiere responder hilo } class VerForo { extension points Si quiere crear hillo Si quiere ver hilo } </pre> <p>The diagram illustrates the use cases for forum subfunctionalities. It features a User actor on the left and four use cases: 'Responder hilo', 'Ver hilo', 'Ver foro', and 'Crear hilo'. A solid line connects the User to each use case. Dashed arrows labeled '<<Extend>>' indicate generalization relationships from 'Responder hilo' to 'Ver hilo', from 'Ver hilo' to 'Ver foro', and from 'Ver hilo' to 'Crear hilo'. The 'Ver hilo' use case contains an 'extension points' section with the condition 'Si quiere responder hilo'. The 'Ver foro' use case contains an 'extension points' section with two conditions: 'Si quiere crear hillo' and 'Si quiere ver hilo'.</p>
Nombre: Ver Foros
Descripción: Permite al usuario acceder al foro, crear hilos nuevos, explorar todos los hilos, además de contestar hilos.
Actores: Usuario
Precondiciones: Estar identificado en el sistema.
Requisitos no funcionales: Ninguno
Flujo de Eventos: <ol style="list-style-type: none"> 1. El usuario selecciona 'Foro' para acceder a él. <ol style="list-style-type: none"> a) Se le muestra toda la lista de hilos. [Figura 3.2.5.2] 1) Si el usuario pulsa en '+Hilo' se le abre una ventana para crear un hilo nuevo. [Figura 3.2.5.3] 2) Si el usuario selecciona un hilo se le mostraran las respuesta al hilo. [Figura 3.2.5.4] a' Si el usuario pulsa en '+Mensaje' se le abre una ventana para responder al hilo [Figura 3.2.5.5]
Poscondiciones: Ninguna
Interfaz Gráfica:

[Library](#) [Home](#) [Catalogue](#) **Foro** [Login](#)

+ HILO

Nombre	Participantes	Mensajes
Hilo1	20	200
Hilo2	20	200
Hilo3	20	200
Hilo4	20	200
Hilo5	20	200
Hilo6	20	200
Hilo7	20	200

Figura 3.2.5.2: Interfaz del foro.

Library Home Catalogue **Foro** Login

Nombre:

Descripcion:

Primer mensaje:

Cancelar

+ HILO

Figura 3.2.5.3: Interfaz para crear un hilo del foro.

Library Home Catalogue **Foro** Login

◀ Hilo1

Ainhize

Javi

+MENSAJE

Figura 3.2.5.4: Interfaz para ver un hilo del foro.

[Library](#) [Home](#) [Catalogue](#) **Foro** [Login](#)

◀ **Hilo1**

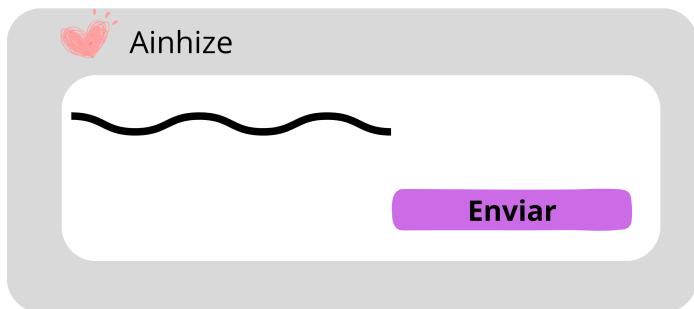


Figura 3.2.5.5: Interfaz para responder a un hilo.

3.2.6. Recomendaciones del sistema

Responsable: Xabier Gabiña
<pre> graph LR User((User)) --> VerCatalogo([Ver Catalogo extension points Si usuario]) VerCatalogo -- "<<Extend>>" --> Recomendaciones([Recomendaciones del sistema]) </pre>
<p>Figura 3.2.6.1: Diagrama de casos de uso de la subfuncionalidad de recomendaciones de libros</p>
Nombre: Recomendaciones del sistema Descripción: Muestra una lista en la parte superior del catálogo mostrando los libros con más préstamos del sistema en función de los gustos del usuario. Actores: Usuario Precondiciones: Estar identificado en el sistema y haber tenido un libro reservado Requisitos no funcionales: Ninguno Flujo de Eventos: 1. El usuario entra en el catalogo para ver los libros disponibles. a) Se le muestra en la parte superior de la pantalla una serie de libros basadas en los generos y peliculas que consume el usuario. [Figura 3.2.6.2]
Poscondiciones: Ninguna Interfaz Gráfica:

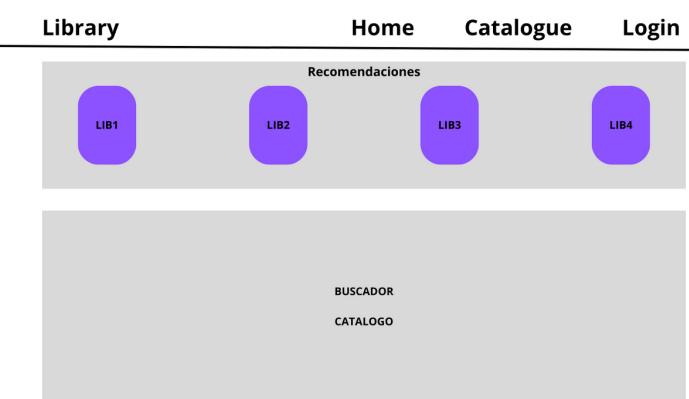


Figura 3.2.6.2: Interfaz de recomendaciones de libros.

3.2.7. Recomendaciones de amigos

Responsable: Kepa Reche												
<pre> graph LR User((User)) --> VerPerfil([Ver Perfil]) VerPerfil --<<include>>--> VerListaAmigos([Ver lista amigos]) VerListaAmigos --<<include>>--> RecComendaciones([Recomendaciones de amigos]) </pre>												
<p>Figura 3.2.7.1: Diagrama de casos de uso de la subfuncionalidad de recomendaciones de amigos</p>												
Nombre: Ver amigos recomendados Descripción: Muestra una lista en la parte superior de la lista de amigos mostrando posibles nuevos amigos basados en amigos de tus amigos o personas con las que tengas lecturas relacionadas. Actores: Usuario Precondiciones: Estar identificado en el sistema y tener algun amigo o libro guardado en tu perfil Requisitos no funcionales: Ninguno Flujo de Eventos: 1. El Usuario entra en su perfil. a) Se le muestra en la parte superior de la lista de amigos varias personas recomendadas segun las lecturas en comun y amigos de amigos. [Figura 3.2.7.2]												
Poscondiciones: Ninguna Interfaz Gráfica: <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #ffd700;">Amigo Recomendado 1</th> <th style="background-color: #ffd700;">Amigo Recomendado 2</th> <th style="background-color: #ffd700;">Amigo Recomendado 3</th> </tr> </thead> <tbody> <tr> <td>Amigo 1</td> <td></td> <td></td> </tr> <tr> <td>Amigo 2</td> <td></td> <td></td> </tr> <tr> <td>Amigo 3</td> <td></td> <td></td> </tr> </tbody> </table> </div>	Amigo Recomendado 1	Amigo Recomendado 2	Amigo Recomendado 3	Amigo 1			Amigo 2			Amigo 3		
Amigo Recomendado 1	Amigo Recomendado 2	Amigo Recomendado 3										
Amigo 1												
Amigo 2												
Amigo 3												

Figura 3.2.7.2: Interfaz de recomendaciones de amigos.

Modelo de Dominio

4.1. Diagrama de modelo de dominio

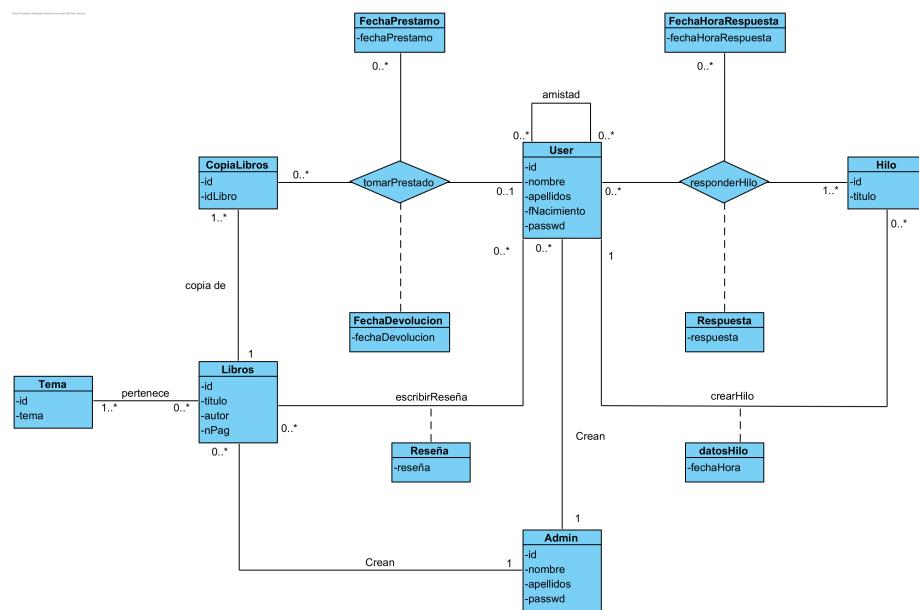


Figura 4.1: Diagrama de modelo de dominio

4.2. Desarrollo del modelo de dominio

4.2.1. Entidades

User

La entidad usuario representa a los usuarios de la aplicación. Dentro almacena sus datos personales como su id, nombre, apellidos, fecha de nacimiento...

Admin

La entidad admin representa a los administradores de la aplicación. Dentro almacena su id, nombre, apellidos y su contraseña.

Libros

La entidad libro representa a los libros de la aplicación. Dentro almacena sus datos como su id, título, autor, número de páginas y género.

CopiaLibros

La entidad copiaLibros representa a las copias de los libros que hay en la aplicación. Dentro almacena su id y el id del libro al que pertenece.

Tema

La entidad tema representa a los temas de los libros. Dentro almacena su id y el nombre del tema.

FechaPrestamo

La entidad fechaPrestamo representa a las fechas de préstamo de los libros. Sirve como clave para diferenciar los diferentes préstamos de un libro que puede hacer un usuario.

Hilo

La entidad hilo representa a los hilos del foro. Dentro almacena la id del hilo y un título.

FechaHoraRespuestas

La entidad fechaHoraRespuesta representa la fecha y hora en la que alguien ha dado una respuesta en el foro y sirve como clave para diferenciar del resto de respuestas que pueda dar un mismo usuario en un mismo hilo.

4.2.2. Relaciones

tomarPrestado

Esta relacion es la que almacena los datos de los libros que se han tomado prestados y lo usuarios que los han tomado. La entidad fechaPrestamo es la encargada de diferencias las dos claves de usuario y libro y permitir asi que un usuario pueda tomar un mismo libro mas de una vez. El atributo de FechaDevolucion unicamente almacena como dato en la relacion la fecha en la que se devuelve el libro.

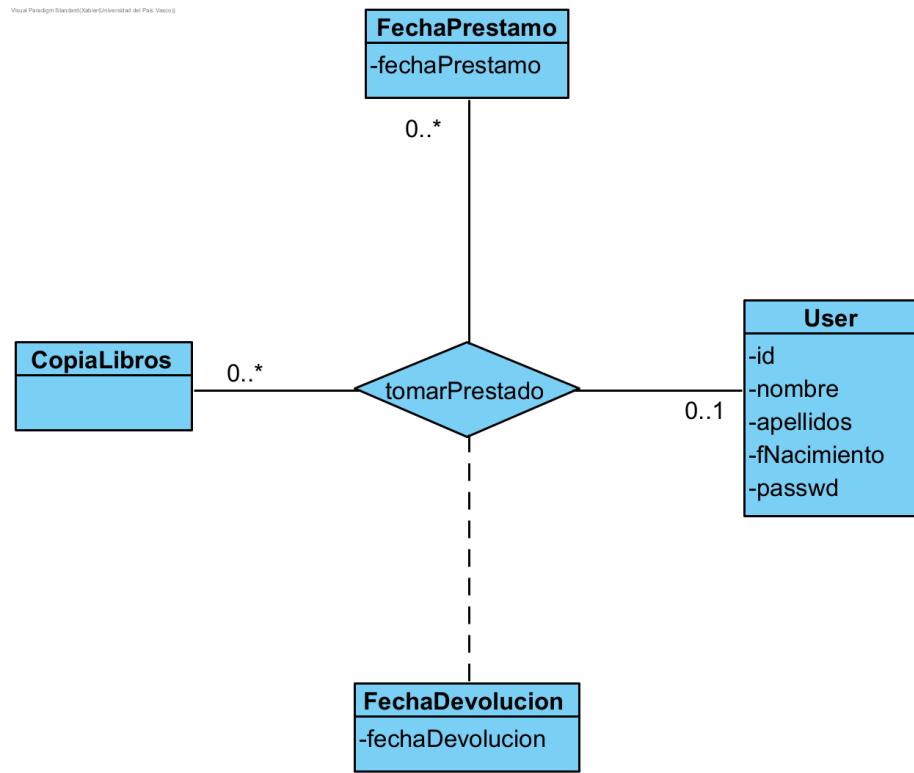


Figura 4.2: Diagrama de modelo de dominio de la relación tomarPrestado

escribirReseña

Esta relación es la que permite guardar la reseña que un usuario pueda escribir en un libro. Al solo tener el atributo Reseña el usuario solo puede escribir una única reseña sobre un libro.

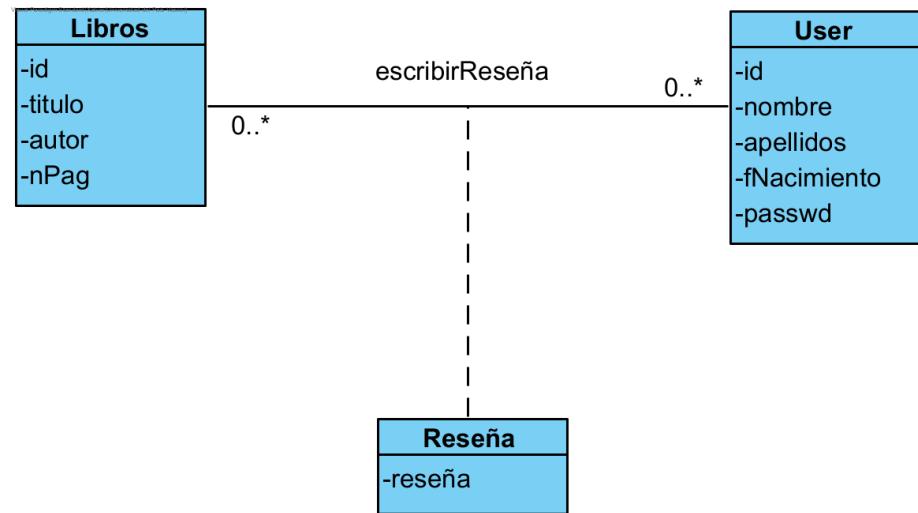


Figura 4.3: Diagrama de modelo de dominio de la relación escribirReseña

amistad

La relacion amistad es la que permite que los usuarios puedan tener amigos en el sistema y guardar dichas amistades.

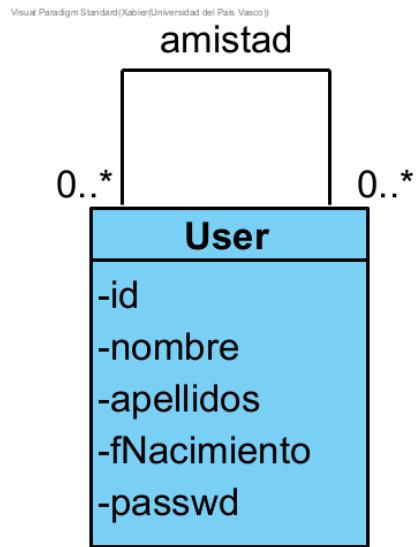


Figura 4.4: Diagrama de modelo de dominio de la relación amistad

responderHilo

Esta relación permite que los usuarios puedan responder a los hilos del foro. La entidad fechaHoraRespuesta es la encargada de diferenciar las respuestas de un mismo usuario en un mismo hilo.

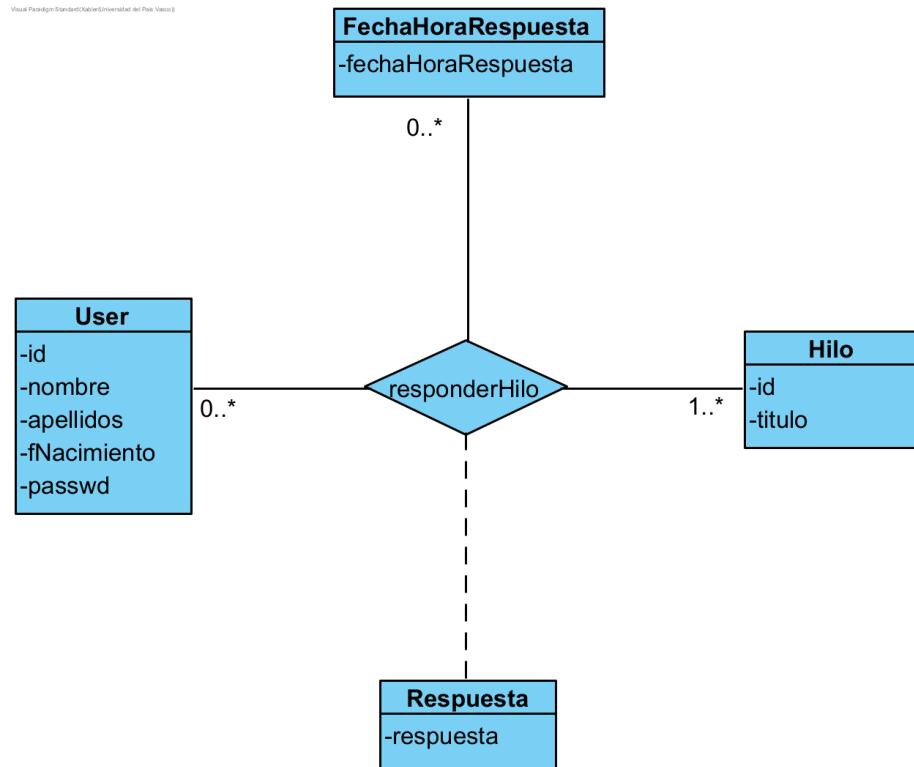


Figura 4.5: Diagrama de modelo de dominio de la relación responderHilo

crearHilo

Esta relacion almacena los datos de creacion de un hilo en el sistema. El sistema guardara el usuario que crea el hilo y en que fecha lo crea. Los hilos son unicos y para evitar repeticiones hacemos que sean unicos por lo que no guardamos la fecha como entidad si no como atributo de la relacion.

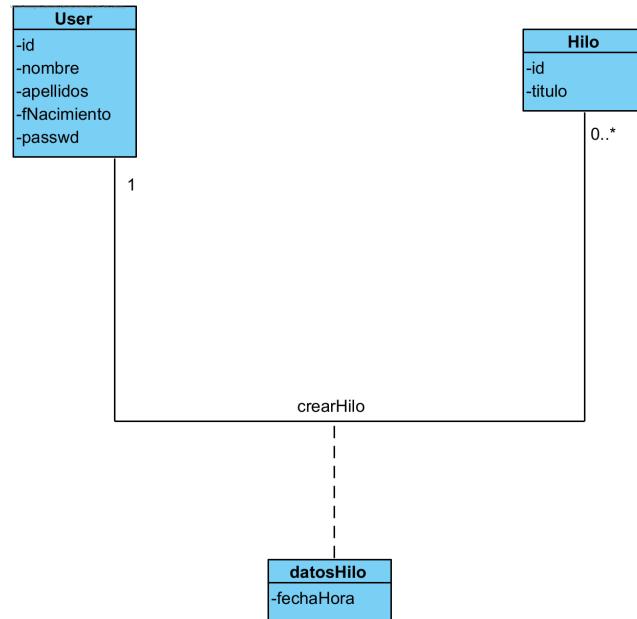


Figura 4.6: Diagrama de modelo de dominio de la relación crearHilo

crean

Esta relacion almacena los datos de creacion de los usuarios y de los libros por parte de un administrados.

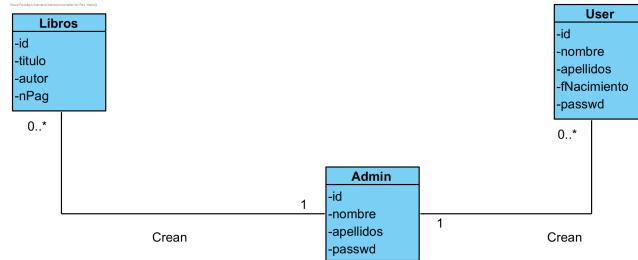


Figura 4.7: Diagrama de modelo de dominio de la relación crean

copiaDe

Esta relación almacena los datos de las copias de los libros que hay en el sistema. La entidad copiaLibros es la encargada de diferenciar las copias de un mismo libro.



Figura 4.8: Diagrama de modelo de dominio de la relación copiaDe

Plan de Pruebas

5.1. Gestión de reservas

Responsable: Kepa Reches Urrutia

Codigo	Descripción	Resultado esperado
F1P1	El usuario selecciona un libro y pulsa reservar	Muestra la interfaz de reserva
F1P1.2	El usuario selecciona un libro y NO pulsa reservar	Se mantiene en la interfaz del catálogo
F1P2	El usuario selecciona una fecha menor de 2 meses	La reserva es correcta
F1P2.2	El usuario selecciona una fecha mayor de 2 meses	Se abre una interfaz de error en la reserva
F1P2.3	El usuario no selecciona ninguna fecha	Se abre una interfaz de error en la reserva

5.2. Reseñas

Responsable: Mohamed El Basri Dehbi

Código	Descripción	Resultado esperado
F2P1	Agregar reseña y puntuación a un libro	La reseña y puntuación se registran correctamente
F2P1.1	Modificar reseña y puntuación de un libro	La reseña y puntuación se actualizan correctamente
F2P2	Intentar agregar reseña y puntuación sin devolver el libro	Se muestra un mensaje de error
F2P3	Ver reseña y puntuación de un libro	La reseña y puntuación del libro son visibles
F2P4	Cambiar reseña y puntuación después de devolver el libro	La reseña y puntuación se pueden cambiar
F2P5	Ver reseña y puntuación de otro usuario	La reseña y puntuación del usuario son visibles
F2P6	Ver reseña y puntuación de un libro sin reseñas	Sistema muestra un mensaje indicando la ausencia de reseñas

5.3. Red de amigos

Responsable: Javier Criado Garcia

Codigo	Descripción	Resultado esperado
F3P1	Ver perfil con sesión iniciada	Muestra el perfil
F3P2	El usuario pulsa el botón “Editar perfil”	Se habilitan los campos para editar el perfil
F3P3	El usuario pulsa el botón “Consultar reservas”	Se abre una interfaz donde ver las reservas
F3P4	El usuario pulsa el botón “Consultar reseñas”	Se abre una interfaz donde ver las reseñas
F3P5	El usuario pulsa el botón “Agregar amigo”	Se abre una interfaz donde añadir un amigo
F3P5.1	El usuario pulsa .°K”	El amigo introducido es añadido
F3P6	El usuario pulsa el botón “Solicitudes de amistad”	Se abre una interfaz donde ver las solicitudes de amistad
F3P7	El usuario pulsa el botón “Borrar amigo”	Se abre una interfaz donde eliminar un amigo
F3P7.1	El usuario pulsa ”SI”	El amigo seleccionado es eliminado
F3P7.2	El usuario pulsa ”NO”	El amigo seleccionado no es eliminado

5.4. Administrador

Responsable: Jon Valdes Granado

Código	Descripción	Resultado esperado
F4P1	El usuario accede como administrador al sistema	Muestra la interfaz de menú del administrador
F4P2	El usuario pulsa el botón “Usuarios”	Muestra la interfaz de gestión de usuario
F4P3	El usuario pulsa el botón “Crear”	Muestra la interfaz de creación de usuario
F4P3.1	El usuario introduce todos los campos correctamente y pulsa el botón “Crear”	Se crea un nuevo usuario correctamente
F4P3.2	El usuario introduce algún campo erróneo o no introduce alguno	Se mostrará un error por pantalla
F4P4	El usuario pulsa el botón “Editar”	Muestra la interfaz de edición de usuario
F4P4.1	El usuario introduce todos los campos correctamente y pulsa el botón “Crear”	Se edita el usuario correctamente
F4P4.2	El usuario introduce algún campo erróneo o no introduce alguno	Se mostrará un error por pantalla
F4P5	El usuario pulsa el botón “Borrar”	Muestra la interfaz de eliminación de usuario
F4P5.1	El usuario introduce el correo electrónico del usuario correctamente y pulsa el botón “Borrar”	Se borra el usuario correctamente
F4P5.2	El usuario introduce un correo electrónico inexistente en el sistema	Se mostrará un error por pantalla

5.5. Foros

Responsable: Janire Veganzones García

Código	Descripción	Resultado esperado
F5P1	El usuario pulsa la opción “Foro” de la cabecera	Se muestra la interfaz principal del foro con todo los hilos creados y con el botón para poder crear uno nuevo
F5P2	El usuario pulsa el botón “+Hilo”	Se muestra la interfaz de crear un hilo
F5P2.1	El usuario pulsa el botón “Cancelar”	No se crea ningún hilo y te redirecciona a la pantalla principal del foro
F5P2.2	El usuario introduce todos los campos correctamente y pulsa el botón “+Hilo”	Se crea un Hilo correctamente y te redirecciona a la pantalla principal del foro donde lo podrás comprobar
F5P2.3	El usuario introduce todos los campos excepto nombre	Se mostrará un mensaje de error por pantalla
F5P2.4	El usuario introduce todos los campos excepto descripción	Se mostrará un mensaje de error por pantalla
F5P2.5	El usuario introduce todos los campos excepto mensaje	Se mostrará un mensaje de error por pantalla
F5P3	EL usuario pulsa sobre el hilo que quiere leer	Se mostrará la interfaz con los mensajes escritos por cada persona hasta ahora y con un botón para poder escribir un mensaje
F5P3.1	El usuario pulsa el botón “+Mensaje”	Se mostrará una interfaz donde podrás escribir tú mensaje para posteriormente publicarlo en el hilo
F5P3.1.1	El usuario escribe un mensaje y pulsa el botón “Enviar”	Te redirecciona a la pantalla anterior donde se pueden ver los mensajes y comprobar que tu mensaje ha sido publicado correctamente

Codigo	Descripción	Resultado esperado
F5P3.1.2	El usuario pulsa el botón .“Enviar” sin escribir ningún mensaje	Se mostrará un mensaje de error por pantalla

5.6. Recomendaciones del sistema

Responsable: Ainhize Martinez Duran

Codigo	Descripción	Resultado esperado
F6.P1	Entrar en el catálogo sin haber iniciado sesión	Muestra el catálogo pero no mostrará ninguna recomendación
F6.P2	Entrar en el catálogo habiendo iniciado sesión pero sin haber hecho ninguna reserva	Muestra el catálogo pero no mostrará ninguna recomendación
F6.P3	Entrar en el catálogo habiendo iniciado sesión y reservado un único libro	Muestra el catálogo y las recomendaciones asociadas al único libro que ha reservado
F6.P3.1	Entrar en el catálogo habiendo iniciado sesión y reservado el único libro que existe de un tema	Muestra el catálogo pero no mostrará ninguna recomendación
F6.P4	Entrar en el catálogo habiendo iniciado sesión y reservado varios libros	Muestra el catálogo y las recomendaciones asociadas a los libros que ha reservado

5.7. Recomendaciones de amigos

Responsable: Xabier Gabiña Barañano

Codigo	Descripción	Resultado esperado
F7P1	Ver perfil sin iniciar sesion	Muestra el perfil sin recomendaciones
F7P1.1	Ver otro perfil al de la sesion	Muestra el otro perfil sin recomendaciones
F7P2	Ver perfil con sesion pero sin amigos ni libros alquilados previamente/actualmente	Muestra el perfil sin recomendaciones
F7P3	Ver perfil con sesion y amigos pero sin libros alquilados previamente/actualmente	Muestra el perfil sin recomendaciones con recomendaciones basadas en los amigos
F7P3.1	Ver perfil con sesion y amigos pero sin libros alquilados previamente/actualmente y tus amigos no tienen otros amigos	Muestra el perfil sin recomendaciones
F7P4	Ver perfil con sesion y libros alquilados previamente/actualmente pero sin amigos	Muestra las recomendaciones en base a los libros alquilados previamente/actualmente
F7P4.1	Ver perfil con sesion y libros alquilados previamente/actualmente pero sin amigos y el libro es el unico de su tema	Muestra el perfil sin recomendaciones
F7P5	Ver perfil con sesion, amigos y libros alquilados previamente/actualmente	Muestra las recomendaciones en base a los libros alquilados previamente/actualmente y los amigos

Diagrama relacional

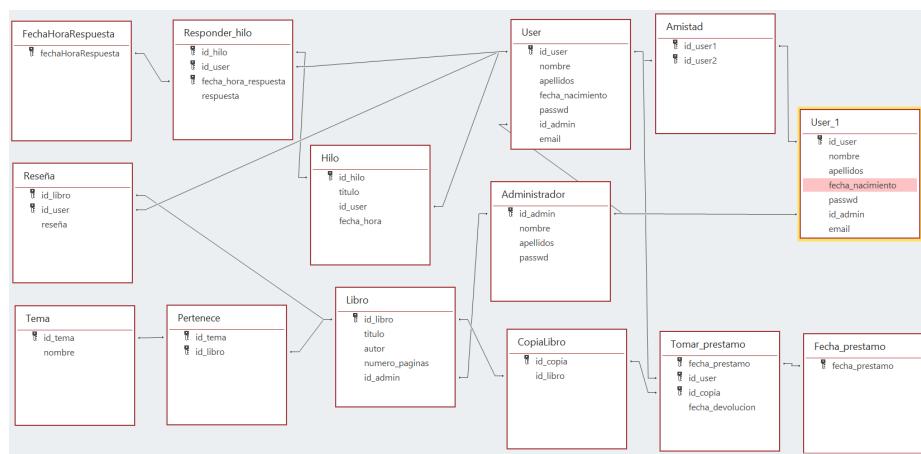


Figura 6.1: Diagrama relacional

Diagrama de clases

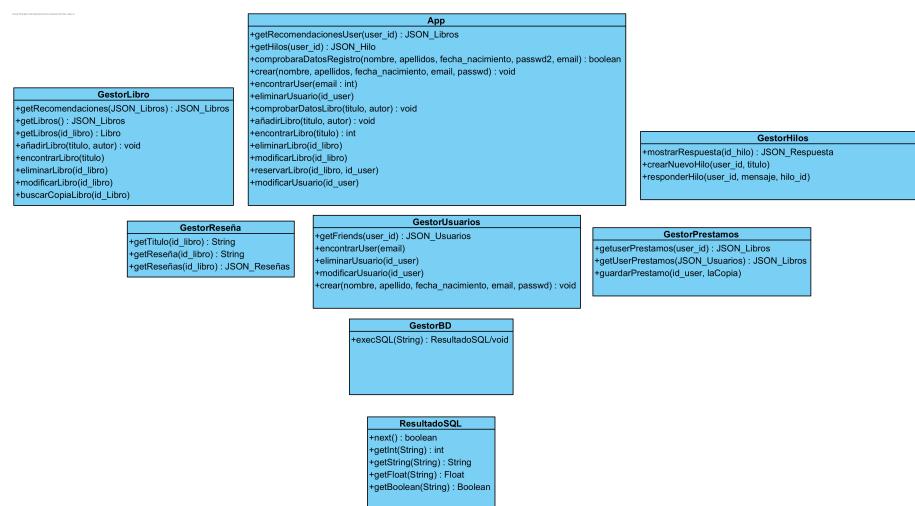


Figura 7.1: Diagrama de clases

Diagramas de comunicación

8.1. Gestión de reservas

Mohamed El Basri

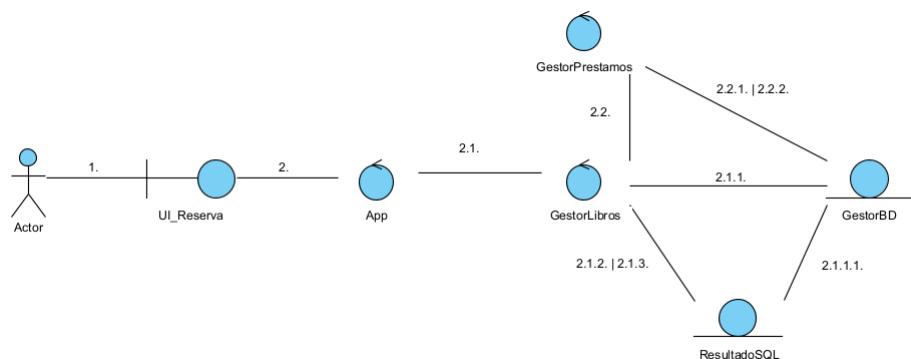


Figura 8.1: Diagrama de comunicación de la gestión de reservas

- 1 Selecciona un libro y pulsa el botón Reservar.
- 2 reserverLibro(id_libro, id_user)
 - 2.1 laCopia=buscarCopiaLibro(id_libro)
 - 2.1.1 execSQL(SELECT id_copia FROM CopiaLibro WHERE id_libro = %id_libro % AND Libre=1) : ResultadoSQL
 - 2.1.1.1 new ResultadoSQL()
 - 2.1.2 next()
 - 2.1.3 getInt(id_copia): int
 - 2.2 guardarPrestamo(id_user, laCopia)
 - 2.2.1 execSQL(INSERT INTO Tomar_prestamo (fecha_prestamo, id_user, id_copia) VALUES (%fecha %, %id_user %, %laCopia %))
 - 2.2.2 execSQL(UPDATE CopiaLibro SET Libre=0 WHERE id_copia= %laCopia %)

8.2. Reseñas

Javier Criado Garcia

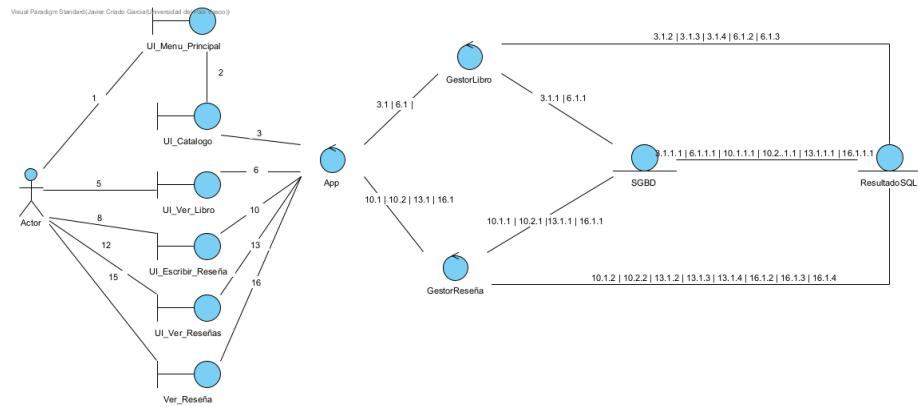


Figura 8.2: Diagrama de comunicación de las reseñas

```

JSON_Libros{
    [
        {
            "id_libro": int,
            "nombre":String
        }
    ]
}
  
```

```

JSON_Reseñas{
    [
        {
            "titulo": String,
            "reseña":String
        }
    ]
}
  
```

1. El usuario selecciona el Catalogo en el menú principal.
2. New UI_CATALOGO(user_id).
3. getLibrosUsuario(user_id):JSON_Libros.

- 3.1 getLibros():JSON.Libros
 - 3.1.1 execSQL(SELECT Libro From Libros)
 - 3.1.1.1 new ResultadoSQL
 - 3.1.2 next()
 - 3.1.3 getInt('id_libro'):int
 - 3.1.4 getString('nombre'):String
- 4. El usuario selecciona un libro.
- 5. New UI_Ver_Libro(user_id, id_libro, nombre).
- 6. getLibro(id_libro): Libro.
 - 6.1 getLibros():JSON.Libros
 - 6.1.1 execSQL(SELECT id_libro FROM Libros WHERE Libro.id_libro= %id_libro %):
 - 6.1.1.1 new ResultadoSQL
 - 6.1.2 next()
 - 6.1.3 getString('nombre'):String
- 7. El usuario selecciona escribir reseña.
- 8. New UI_Escribir_Reseña(user_id, id_libro, nombre).
- 9. El usuario pulsa el botón enviar.
- 10. EscribirReseña(user_id):void.
 - 10.1 elTitulo: getTitulo(): String
 - 10.1.1 execSQL(SELECT id_libro FROM Libros WHERE Libro.id_libro= %id_libro %):
 - 10.1.1.1 new ResultadoSQL
 - 10.1.2 next()
 - 10.2 laReseña: getReseña(id_libro):String
 - 10.2.1 execSQL(INSERT INTO Reseñas.reseña SELECT laReseña where User.id_user= %user_id AND Libro.id_libro= %id_libro %): ResultadoSQL
 - 10.2.1.1 new ResultadoSQL
 - 10.2.2 next()
- 11. El usuario selecciona ver reseñas
- 12. New UI_Ver_Reseñas(id_libro)
- 13. getReseñas(id_Libro): JSONReseñas
 - 13.1 getReseña(id_libro):String

- 13.1.1 execSQL(SELECT Reseña From Reseñas where Libro.id_libro= %id_libro %): ResultadoSQL
 - 16.1.1.1 new ResultadoSQL
 - 13.1.2 next()
 - 13.1.3 getTitulo(' id_titulo'):String
 - 13.1.4 getReseña('id_reseña'):String
14. El usuario selecciona una reseña
15. New UI_Ver_Reseña(id_libro,id_reseña)
16. getReseña():JSONReseña
 - 16.1 getReseña(id_libro):String
 - 16.1.1 execSQL(SELECT Reseña From Reseñas where Libro.id_libro= %id_libro % AND Reseña.reseña= %id_reseña %): ResultadoSQL
 - 16.1.1.1 new ResultadoSQL
 - 16.1.2 next()
 - 16.1.3 getTitulo(' id_titulo'):String
 - 16.1.4 getReseña('id_reseña'):String

8.3. Red de amigos

Jon Valdes Granado

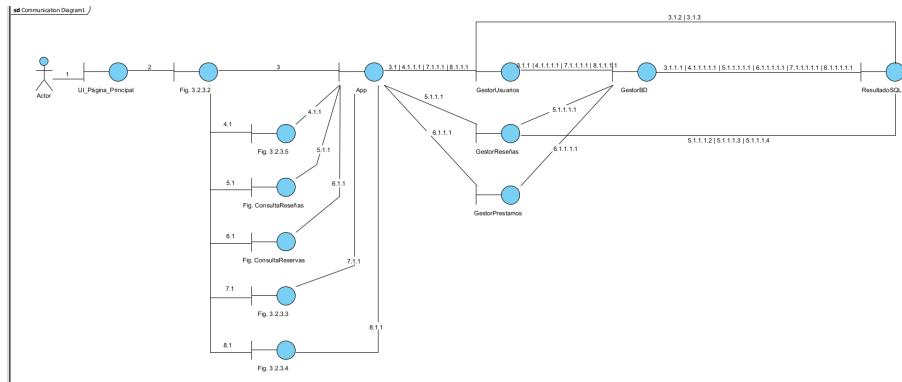


Figura 8.3: Diagrama de comunicación de la red de amigos

```

JSON_Libros{
    [
        {
            "id_libro":int,
            "nombre":String
        }
    ]
}

JSON_Usuarios{
    [
        {
            "id_user": int
        }
    ]
}

JSON_Reseñas{
    [
        {
            "titulo": String,
            "reseña": String
        }
    ]
}

```

- 1 El usuario selecciona el perfil en el menú principal.
- 2 new UIPerfil(*user_id*).getAmigos(*user_id*) : JSON_Usuarios
- 3 3.1 listaAmigos = getFriendsFriends(*user_id*):JSON_Usuarios
 - 3.1.1 execSQL(" execSQL(" SELECT DISTINCT a2.usuario2 AS amigo FROM amigos a1 JOIN amigos a2 ON a1.usuario2 = a2.usuario1 WHERE a1.usuario1 = a1.id_user AND a2.usuario2 = '%user_id %'; "): new ResultadoSQL()
 - 3.1.1.1 new ResultadoSQL()
 - 3.1.2 next()
 - 3.1.3 getString('nombre'):String

Editar perfil

- 4.1 new Fig. 3.2.3.5.
 - 4.1.1 El usuario introduce los datos del usuario
 - 4.1.1.1 modificarDatos(*id_user*)
 - 4.1.1.1.1 modificarDatos(*id_user*)
 - 4.1.1.1.1.1 execSQL(UPDATE User SET nombre = %Nuevo nombre%, apellidos = %Nuevos apellidos%, fecha_nacimiento = %Nueva fecha nacimiento%, passwd = %Nueva contraseña% WHERE id_user = *id_user*)

Consultar reseñas

- 5.1 new Fig. (*id_user*)
 - 5.1.1 getReseñas(*id_user*) : JSON_Reseñas
 - 5.1.1.1 getReseña(*id_user*) : Reseña
 - 5.1.1.1.1 execSQL((SELECT id_libro FROM Libros WHERE Libro.id_libro= ResultadoSQL)

5.1.1.1.1 new ResultadoSQL

next()
 getTitulo('idLibro'):String
 getReseña('idLibro'):String

Consultar reservas

- 6.1 new Fig. (*id_user*)
 - 6.1.1 getUserPrestamos(*id_user*) : JSON_Reservas
 - 6.1.1.1 getUserPrestamo(*id_user*) : JSON_Reservas
 - 6.1.1.1.1 execSQL((SELECT id_copia, fechaPrestamo FROM Prestamo WHERE id_user = ResultadoSQL)

6.1.1.1.1 new ResultadoSQL

```
next()  
getIdCopia('idLibro'):int  
getfechaPrestamo('idLibro'):Date  
Añadir amigo
```

7.1 new Fig.3233 (id_{user})

7.1.1 añadirAmigo($\text{user}_i d$)

7.1.1.1 execSQL((INSERT INTO Amigo VALUES user_I) $D = \text{ResultadoSQL}$)

Eliminar amigo

7.1 new Fig.3233 (id_{user})

7.1.1 añadirAmigo($\text{user}_i d$)

7.1.1.1 execSQL((DELETE INTO Amigo VALUES user_I) $D = \text{ResultadoSQL}$)

8.4. Red de amigos

8.5. Administrador

Janire Veganzones García

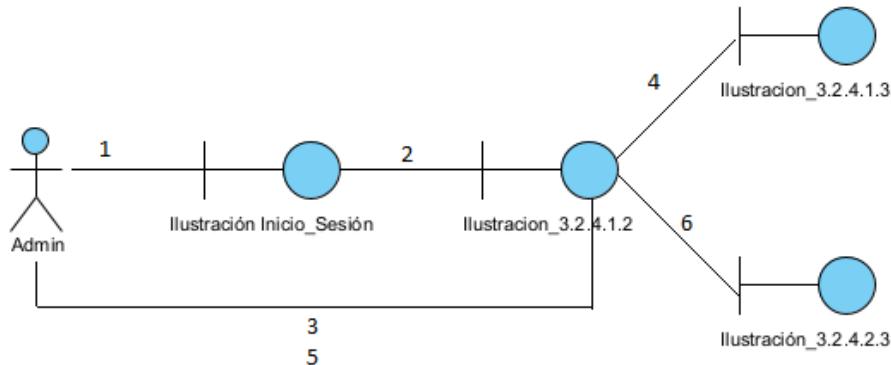


Figura 8.4: Diagrama de comunicación general del administrador

1. El usuario inicia sesión como Admin.
2. new illustration_3.2.4.1.2().
3. Pulsa “Usuarios”.
4. new illustration_3.2.4.1.3().
5. Pulsa “Libros”.
6. new illustration_3.2.4.2.3().

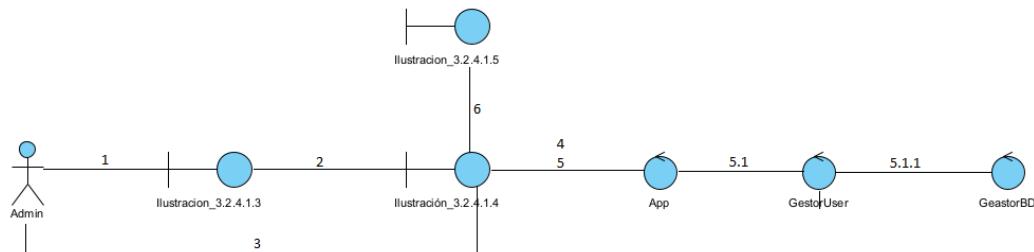


Figura 8.5: Diagrama de comunicación Crear un Usuario

1. El usuario pulsa “Crear”.
2. new `ilustracion_3.2.4.1.3()`.
3. Introducir nombre, apellidos, fecha nacimiento, email, contraseña y contraseña2 y pulsar “Crear”.
4. `comprobarDatosRegistro(nombre, apellidos, fecha_nacimiento, passwd, passwd2, email) : boolean`
[Si `comprobarDatosRegistro == TRUE`]
5. `crear(nombre, apellidos, fecha_nacimiento, email, passwd): void`
 - 5.1 `crear(nombre, apellidos, fecha_nacimiento, email, passwd): void`
 - 5.1.1 `execSQL (INSERT INTO User VALUES (%nombre %, %apellidos %, %fecha_nacimiento %, %email %, %passwd %))`: ResultadoSQL
[Si No]
6. new `ilustracion_3.2.4.1.5()`.

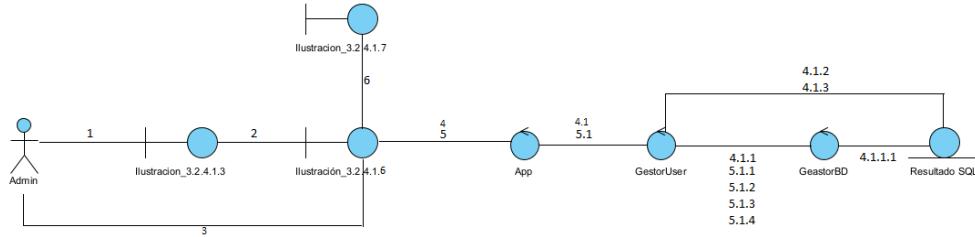


Figura 8.6: Diagrama de comunicación Borrar un Usuario

1. El usuario pulsa “Borrar”.
2. new illustration_3.2.4.1.6().
3. Escribir el email del usuario que deseas eliminar y pulsar “Borrar”.
4. usuarioBuscado = encontrarUser(email): int
 - 4.1 encontrarUser(email)
 - 4.1.1 execSQL(SELECT id_user FROM User WHERE email = %email %): ResultadoSQL
 - 4.1.1.1 new ResultadoSQL()
 - 4.1.2 next()
 - 4.1.3 getInt(id_user): int
 [SI usuarioBuscado != NULL]
 5. eliminarUsuario(id_user)
 - 5.1 eliminarUsuario(id_user)
 - 5.1.1 execSQL(DELETE * FROM Reseña WHERE id_user= usuarioBuscado)
 - 5.1.2 execSQL(DELETE * FROM Responder_hilo WHERE id_user= usuarioBuscado)
 - 5.1.3 execSQL(DELETE * FROM Tomar_prestamo WHERE id_user = usuarioBuscado)
 - 5.1.4 execSQL(DELETE * FROM User WHERE id_user= UsuarioAElimiar)
 - [Si No]
 6. new illustration_3.2.4.1.7().

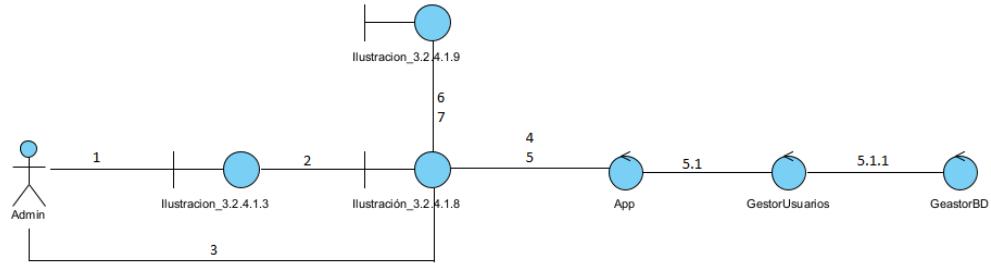


Figura 8.7: Diagrama de comunicación Modificar un Usuario

1. El usuario pulsa “Modificar”.
2. new illustration_3.2.4.1.3().
3. Introducir Nombre, Apellidos, Fecha nacimiento, Email, Nuevo nombre, Nuevos Apellidos, Nueva fecha nacimiento, Nueva contraseña, Repetir nueva contraseña y pulsar “Modificar”.
4. encontrarUser(email) [Función desarrollada en el diagrama anterior]
[SI usuarioBuscado != NULL]
[SI %Nueva contraseña% == %Repetir nueva contraseña%]
5. modificarUsuario(id_user)
 - 5.1 modificarUsuario(id_user)
 - 5.1.1 execSQL(UPDATE User SET nombre = %Nuevo nombre%, apellidos = %Nuevos apellidos%, fecha_nacimiento = %Nueva fecha nacimiento%, passwd = %Nueva contraseña% WHERE id_user = id_user)
 - [Si No]
 6. new illustration_3.2.4.1.9().
[Si No]
 7. new illustration_3.2.4.1.9().

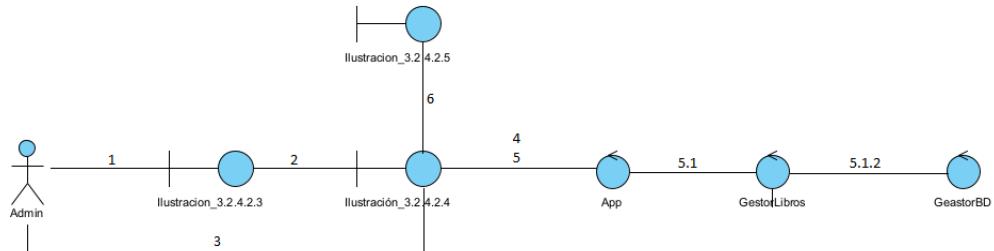


Figura 8.8: Diagrama de comunicación Insertar un Libro

1. El usuario pulsa “Añadir”.
2. new ilustration_3.2.4.2.3().
3. Rellenar los campos Título y autor y pulsar “Insertar”.
4. comprobarDatosLibro(título, autor): boolean
[Si comprobarDatosLibro == TRUE]
5. añadirLibro(título,autor): void
- 5.1 añadirLibro(título,autor): void
5.1.1 execSQL (INSERT INTO Libros (%título%,%autor%)): ResultadoSQL
[Si No]
6. new ilustration_3.2.4.2.5().

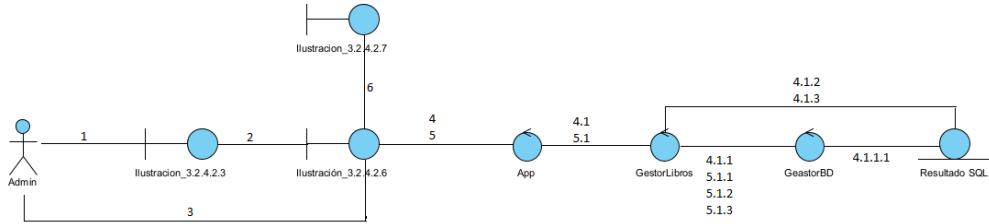


Figura 8.9: Diagrama de comunicación Borrar un Libro

1. El usuario pulsa “Eliminar”.
2. new ilustration_3.2.4.2.6().
3. Escribir el titulo y autor del libro que deseas eliminar y pulsar “Eliminar”.
4. libroBuscado= encontrarLibro(titulo): int
 - 4.1 encontrarLibro(titulo)
 - 4.1.1 execSQL(SELECT id.libro FROM Libro WHERE titulo = %titulo %): ResultadoSQL
 - 4.1.1.1 new ResultadoSQL()
 - 4.1.2 next()
 - 4.1.3 getInt(id.libro): int

[SI libroBuscado != NULL]
 5. eliminarLibro(id.libro)
 - 5.1 eliminarLibro(id.libro)
 - 5.1.1 execSQL(DELETE * FROM Reseña WHERE id.libro= libroBuscado)
 - 5.1.2 execSQL(DELETE * FROM CopiaLibro WHERE id.libro= libroBuscado)
 - 5.1.3 execSQL(DELETE * FROM Libro WHERE id.libro= libroBuscado)

[Si No]
 6. new ilustration_3.2.4.2.7().

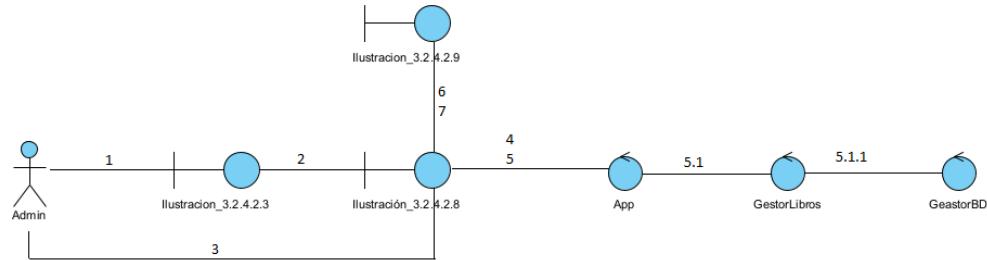


Figura 8.10: Diagrama de comunicación Modificar un Libro

1. El usuario pulsa “Modificar”.
2. new `ilustration_3.2.4.2.8()`.
3. Introducir titulo anterior, autor anterior, titulo nuevo y autor nuevo y pulsar “Modificar” .
4. encontrarLibro(titulo) [Función desarrollada en el diagrama anterior]
[SI libroBuscado != NULL]
5. modificarLibro(id_libro)
 - 5.1 modificarLibro(id_libro)
 - 5.1.1 execSQL(UPDATE Libro SET titulo=%titulo nuevo%, autor=%autor nuevo%)

[Si No]
6. new `ilustration_3.2.4.2.9()`.
[Si No]
7. new `ilustration_3.2.4.2.9()`.

8.6. Foros

Ainhize Martinez Duran

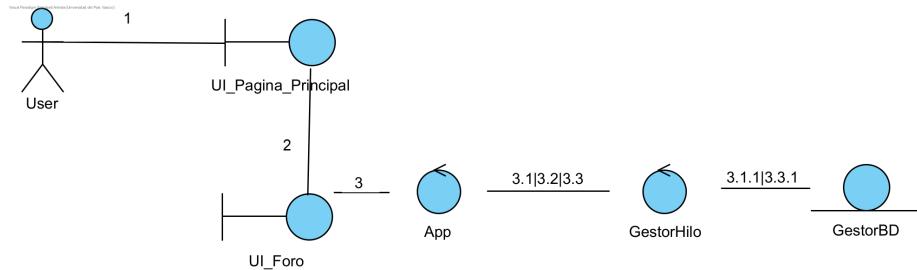


Figura 8.11: Diagrama de comunicación para Foro

```
JSON_Hilo{  
    [  
        {  
            "id_hilo":int,  
            "titulo":String  
        }  
    ]  
}
```

```
JSON_Respuesta{  
    [  
        {  
            "mensaje":String,  
            "user":String  
        }  
    ]  
}
```

1. El usuario selecciona Foro en el menú principal.
2. new UI_Foro(user_id)
3. getHilos(user_id):JSON_Hilo
 - 3.1 crearNuevoHilo(user_id,titulo)
 - 3.1.1 execSQL(INSERT INTO Hilo (id_user,titulo) VALUES(%user_id %, %titulo %))
 - 3.2 mostrarRespuesta(id_hilo):JSON_Respuesta

3.3 responderHilo(user_id,mensaje,hilo_id)

3.3.1 execSQL(INSERT INTO Responder_Hilo (id_user,respuesta,id_hilo)
VALUES (%user_id %, %mensaje %, %hilo_id %))

8.7. Recomendaciones del sistema

Xabier Gabiña Barañano

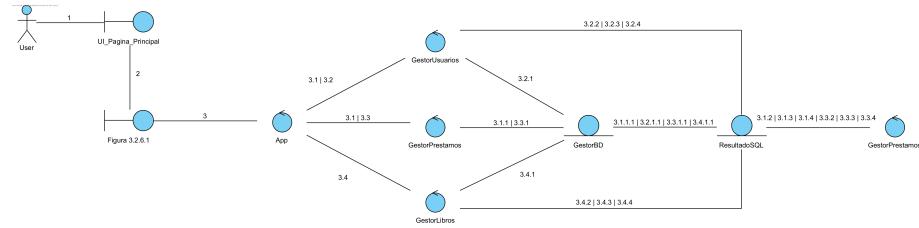


Figura 8.12: Diagrama de comunicación de las recomendaciones del sistema

```
JSON_Libros{  
    [  
        {  
            "id_libro":int,  
            "nombre":String  
        }  
    ]  
}  
  
JSON_Usuarios{  
    [  
        {  
            "id_user":int  
        }  
    ]  
}
```

- 1 El usuario selecciona el catálogo en el menú principal.
- 2 new Figura_3.2.6.1(user_id).
- 3 getRecomendacionesUser(user_id):JSON_Libros
 - 3.1 listaLibros=getUserPrestamos(user_id):JSON_Libros
 - 3.1.1 execSQL(" SELECT Libro.* FROM Tomar_prestamo INNER JOIN CopiaLibro ON Tomar_prestamo.id_copia = CopiaLibro.id_copia INNER JOIN Libro ON CopiaLibro.id_libro = Libro.id_libro WHERE Tomar_prestamo.id_user = %user_id %; "): new ResultadoSQL()
 - 3.1.1.1 new ResultadoSQL()
 - 3.1.2 next()
 - 3.1.3 getInt('id_libro'):int
 - 3.1.4 getString('nombre'):String
 - 3.2 listaAmigos=getFriends(user_id):JSON_Usuarios
 - 3.2.1 execSQL(" SELECT U1.id_user AS id_amigo, U1.nombre AS nombre_amigo FROM User AS U JOIN Amistad AS A ON U.id_user = A.id_user1 OR U.id_user = A.id_user2 JOIN User AS U1 ON (U.id_user = A.id_user1 AND U1.id_user = A.id_user2) OR (U.id_user = A.id_user2 AND U1.id_user = A.id_user1) WHERE U.id_user = %user_id %; "): new ResultadoSQL()
 - 3.2.1.1 new ResultadoSQL()
 - 3.2.2 next()
 - 3.2.3 getInt('id_amigo'):int
 - 3.3 listaLibros+=getUserPrestamos(listaAmigos):JSON_Libros
 - 3.3.1 execSQL(" SELECT Libro.* FROM Tomar_prestamo INNER JOIN CopiaLibro ON Tomar_prestamo.id_copia = CopiaLibro.id_copia INNER JOIN Libro ON CopiaLibro.id_libro = Libro.id_libro WHERE Tomar_prestamo.id_user = %user_id %; "): new ResultadoSQL()
 - 3.3.1.1 new ResultadoSQL()
 - 3.3.2 next()

3.3.3 getInt('id_libro'):int

3.3.4 getString('nombre'):String

3.4 getRecomendaciones(listaLibros):JSON_Libros

3.4.1 execSQL(" SELECT Libro.* FROM Libro INNER JOIN Pertenece ON Libro.id_libro = Pertenece.id_libro INNER JOIN Tema ON Pertenece.id_tema = Tema.id_tema WHERE Tema.id_tema IN (SELECT Pertenece.id_tema FROM Pertenece WHERE Pertenece.id_libro = %libro_id %); "): new ResultadoSQL()

3.4.1.1 new ResultadoSQL()

3.4.2 next()

3.4.3 getInt('id_libro'):int

3.4.4 getString('nombre'):String

8.8. Recomendaciones de amigos

Kepa Reche Urrutia

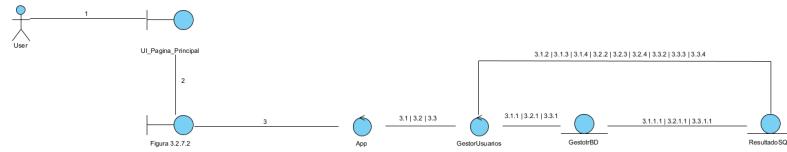


Figura 8.13: Diagrama de comunicación de las recomendaciones de amigos

```
JSON_Libros{  
    [  
        {  
            "id_libro":int,  
            "nombre":String  
        }  
    ]  
}  
JSON_Usuarios{  
    [  
        {  
            "user_id":int,  
            "nombre":String  
        }  
    ]  
}
```

1. El usuario selecciona el perfil en el menú principal.
2. new UI_PERFIL(user_id).
3. getRecomendacionesAmigos(user_id):JSON_Usuarios
 - 3.1 listaLibros=getLibros():JSON_Libros
 - 3.1.1 execSQL(" SELECT Libro.* FROM User INNER JOIN Tomar_prestamo ON User.id_user = Tomar_prestamo.id_user INNER JOIN CopiaLibro ON Tomar_prestamo.id_copia = CopiaLibro.id_copia INNER JOIN Libro ON CopiaLibro.id_libro = Libro.id_libro WHERE User.id_user = %user_id% "); ResultadoSQL
 - 3.1.1.1 new ResultadoSQL()
 - 3.1.2 next()
 - 3.1.3 getInt('id_libro'):int
 - 3.1.4 getString('nombre'):String
 - 3.2 listaUsuarios=getFriendsFriends(user_id):JSON_Usuarios
 - 3.2.1 execSQL(" SELECT DISTINCT a2.usuario2 AS amigodeamigo FROM amigos a1 JOIN amigos a2 ON a1.usuario2 = a2.usuario1 WHERE a1.usuario1 = a1.id_user AND a2.usuario2 = '%user_id%');
 - 3.2.1.1 new ResultadoSQL()
 - 3.2.2 next()
 - 3.2.3 getInt('user_id'):int
 - 3.2.4 getString('nombre'):String
 - 3.3 getAmigosPorRecomendaciones(listaLibros):JSON_Usuarios
 - 3.3.1 execSQL(" SELECT DISTINCT a2.usuario2 AS amigodeamigo, l.id_libro FROM amigos a1 JOIN amigos a2 ON a1.usuario2 = a2.usuario1 JOIN lecturas l ON a2.usuario2 = l.user_id WHERE a1.usuario1 = 'user_id' AND a2.usuario2 = 'user_id' "); ResultadoSQL
 - 3.3.1.1 new ResultadoSQL()
 - 3.3.2 next()
 - 3.3.3 getInt('user_id'):int
 - 3.3.4 getString('nombre'):String