GLIB

Alumno: Miguel Borja Abril Zamarro
Grado en Ingenería Informática

Consultor: Albert Grau Perisé

13/06/2017

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento 3.0 España de Creative Commons

B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	GLIB
Nombre del autor:	Miguel Borja Abril Zamarro
Nombre del consultor:	Albert Grau Perisé
Fecha de entrega (mm/aaaa):	13/06/2017
Área del Trabajo Final:	J2EE
Titulación:	*Grado en Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

El objetivo de este Trabajo Final de Grado de desarrollo en tecnologías J2EE es, aparte de aplicar conocimientos en las tecnologías J2EE, aplicar las competencias adquiridas a lo largo de los estudios en el Grado en Ingeniería Informática: Gestión de Proyectos, Técnicas de Desarrollo de Software, Ingeniería de Requisitos, Diseño de bases de datos, entre otras.

Se han aplicado las siguientes tecnologías J2EE: JSF (Java Server Faces) en la capa de presentación, ManagedBean en la capa de negocio, y JPA en la capa de integración. Se han mejorado estas tecnologías con el framework PrimeFaces, permitiendo que la interfaz sea más amigable para el usuario y ofreciendo capacidades que JSF no dispone.

La aplicación web implementa la gestión diaria del trabajo en una librería, incluyendo el alta de autores y libros, el alta de almacenes, encargar libros a los almacenes, reservar libros a clientes, generación y edición de facturas, siendo estas las principales funcionalidades.

Abstract (in English, 250 words or less):

The objective of this Final Degree Work focused on developing applications based on J2EE technologies, besides from implementing J2EE technologies, apply the competencies acquired during the subjects coursed: Project Management, Software Development Techniques, Requirement engineering, Database design, amongst others.

To implement the application the J2EE technologies applied are: JSF (Java Server Faces) in the presentation layer, Managed Bean in the business layer, and JPA in the integration layer. These technologies have been hardened with the PrimeFaces framework, making the interface more user-friendly and offering possibilities JSF does not.

The web application implements the daily management of a library, including the creation and edition of authors and books, the creation of warehouses, order books to warehouses, reserve books, generate and edit bills. These functionalities are the main ones.

Palabras clave (entre 4 y 8):
J2EE, PrimeFaces, Scrum, Gestión de Proyectos

Índice

1. Introducción

1.1 Contexto y justificación del Trabajo

En este TFG la necesidad a cubrir es desarrollar una aplicación que permite la gestión del trabajo diario en una librería, incluyendo, la edición y la creación de autores y de libros, el alta y la edición de almacenes, realizar pedidos (encargos) a los almacenes, reservar libros, generar facturas, editar las preferencias de autores y temáticas a los clientes, edición del perfil de usuario, constituyendo este conjunto de funcionalidades el núcleo de la aplicación.

Es un tema relevante porque permite desarrollar una aplicación basada en tecnologías J2EE, logrando de esta manera aplicar los objetivos del TFG en cuanto a tecnología, que proporcionan un marco de trabajo y unas características adecuadas para aplicaciones de entorno empresarial, respondiendo tanto a requisitos funcionales como no funcionales. En cuanto a la aplicación de las competencias adquiridas en el Grado en Ingeniería Informática es relevante porque constituye un proyecto con el que se pueden aplicar dichas competencias.

Se resuelve la situación documentado en primer lugar los requisitos de la aplicación a desarrollar, en segundo lugar los casos de uso de la aplicación, en tercer lugar diseñando el modelo a implementar en el desarrollo de la aplicación teniendo en cuenta que la tecnología será J2EE, en cuarto lugar diseñando el modelo de datos (aplicando conocimientos de diseño de bases de datos, y en quinto lugar diseñando el prototipo de la aplicación (la interfaz). A partir de esta documentación, se puede implementar la aplicación con una metodología de diseño clara, y teniendo en cuenta qué se quiere obtener.
Adicionalmente, se aplican metodologías de desarrollo como metodologías ágiles (SCRUM) que permite disponer siempre de un producto operativo, y ajustarse a posibles desviaciones en la planificación inicial del proyecto.

El objetivo es obtener una aplicación que permite a una librería gestionar el trabajo diario, cumpliendo los requisitos obtenidos en la fase de obtención de requisitos.

1.2 Objetivos del Trabajo

1.2.1 Objetivos generales

- Permitir a los empleados de la librería la gestión de la misma mediante una aplicación web, sin tener que depender de infraestructura hardware y software instalados en cada ordenador, disponiendo de una web en un servidor con su persistencia en una base de datos alojada en otro servidor.

- Permitir a los clientes interactuar con la librería y sus empleados a través de la web, sin necesidad de desplazarse hasta la librería para realizar gestiones, excepto para recoger libros, ya que inicialmente no se envían libros a domicilio, con el objetivo de conocer personalmente a los clientes, y que puedan visitar la librería.

1.2.2 Objetivos específicos

- Desarrollar en tecnologías de la plataforma J2EE una aplicación web mediante JSF, Managed Bean y JPA, reforzado con el framework PrimeFaces, proporcionando una interfaz gráfica sencilla, intuitiva y amigable.

1.2.3 Objetivos didácticos

Aprendizaje o profundización en las siguientes tecnologías de desarrollo de software:

- Lenguaje de programación Java, de la versión 1.8.
- *J2EE*
    - *JSF (Java Server Faces)*: lenguaje de marcado que permite implementar la interfaz de la aplicación web, a la vez que interactuar con los Managed Bean (objetos que recogen la interacción del usuario con la interfaz y los transmiten a la capa de negocio). JSF pertenece a la capa de presentación.
    - *Managed Bean*: son clases Java de la capa de presentación, que a través de anotaciones son accesibles desde las JSF, siendo su objetivo principal recoger la interacción del usuario con las JSF, y transmitirlas a la capa de negocio (EJB).
    - *EJB*: son clases e interfaces Java, de la capa de negocio, que reciben (desde Managed Bean) las peticiones que realiza el usuario a través de la capa de presentación, tratando los datos recibidos, logrando que sean persistentes en base de datos, y retornando una respuesta (satisfactoria o no, al usuario, a través de la capa de presentación.
    - *JPA*: es un marco de trabajo que permite representar entidades mediante clases Java, correspondientes a las tablas de base de datos, logrando persistencia de los datos con que trabaja la aplicación a través de las entidades y el lenguaje de consultas *JPQL*. Esta capa representa la capa de integración.
- *PrimeFaces*: es un marco de trabajo que se integra con JSF haciendo la interfaz más amigable al usuario y proporcionando características que JSF no dispone.
- *MySQL*: sistema gestor de bases de datos que se aplica para persistir los datos con que trabaja la aplicación.
- Aplicar los conocimientos necesarios que se hayan adquirido a lo largo de las asignaturas cursadas durante los estudios del Grado en Ingeniería Informática.

1.3 Enfoque y método seguido

La estrategia elegida para implementar el TFG ha sido desarrollar un producto nuevo, ya que de esta manera se profundiza mejor en el aprendizaje de las tecnologías mencionadas en el apartado anterior, y

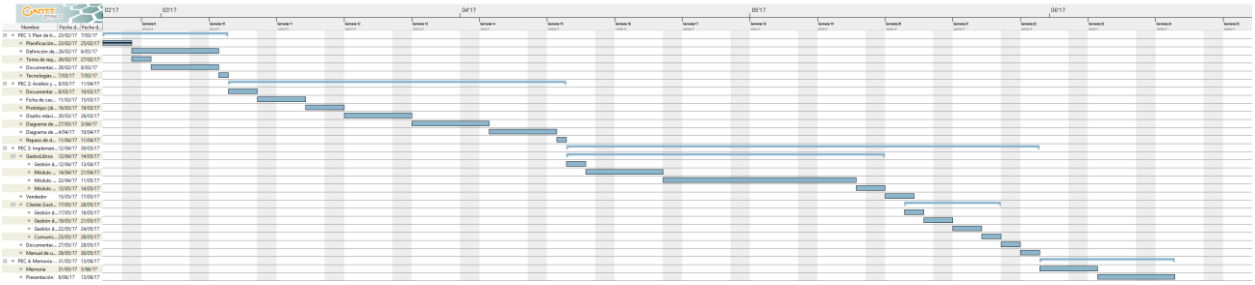
debido a que el cliente requiere de una aplicación a medida que satisfaga sus requisitos funcionales concretos.

1.4 Planificación del Trabajo

Los recursos necesarios para llevar a cabo el TFG son:

- PC de sobremesa o portátil.
- Entorno de desarrollo (IDE): Eclipse (versión NEON).
- Tecnologías *J2EE: JSF, Managed Bean* y *JPA*.
- Sistema gestor de base de datos: *MySQL*.
- *PrimeFaces*: descarga de la librería versión Community, para añadirla al proyecto, y consulta de la documentación oficial.
- Servidor de aplicaciones: Wildfly 10.1.0 Final.
- MySQL (servidor de base de datos) y MySQL Workbench (aplicación de escritorio para administrar el servidor).
- *GanttProject*: herramienta para generar la planificación del proyecto.
- *MagicDraw*: herramienta para documentar el diseño de la aplicación (diseño de las capas, diseño de la base de datos).

A continuación se incluye una imagen con la planificación temporal del TFG:



1.5 Breve sumario de productos obtenidos

Los productos obtenidos durante el desarrollo de la aplicación son:

- Plan de trabajo: objetivos del proyecto, requisitos funcionales de la aplicación, planificación temporal del proyecto, tecnologías propuestas y metodologías de desarrollo de software.
- Documentación de análisis y diseño: descripción textual y diagramas de casos de uso, diseño de la interfaz de la aplicación (prototipo), diseño relacional de la base de datos, diagrama de clases y diagrama de arquitectura.
- Proyecto Java (Eclipse): representa la implementación de la aplicación, el código fuente.
- Documento de pruebas: Documenta las pruebas funcionales de la aplicación web.
- Manual de despliegue: documenta cómo implementar el entorno de desarrollo de la aplicación.

- Presentación virtual: vídeo explicativo y demostrativo de la aplicación web GLIB.
- Memoria: el presente documento.

1.6 Breve descripción de los otros capítulos de la memoria

En este apartado se describe el contenido del resto de capítulos de la memoria:

- Análisis: en este apartado se describen las necesidades del proyecto, los requisitos funcionales y no funcionales, así como el diseño conceptual. Su relación con el trabajo global es que este capítulo constituye la base para poder desarrollar el contenido del resto del trabajo, es aquello que el cliente espera de la aplicación web.
- Diseño: en este capítulo se diseña la aplicación, como fase previa a la implementación, comenzando con la arquitectura de la aplicación (en 3 capas), progresando al diseño estructurado en componentes e independiente de la tecnología, para culminar en el diseño en componentes adaptado a la tecnología concreta: J2EE. Así mismo, se diseñan los diagramas de secuencia de aquellos casos de uso que se consideran más complejos, y se genera el diseño relacional de la base de datos, para implementarlo en la siguiente fase, al definir las entidades JPA. Por último se diseña el prototipo de la aplicación, en cuanto a la interfaz gráfica del usuario. Su relación con el trabajo global radica en que el diseño, a partir del análisis inicial, documenta la estructura de la aplicación, desde la arquitectura a nivel global, hasta el diseño detallado y dependiente de la arquitectura y la tecnología escogidas, permitiendo continuar con la implementación.
- Implementación: en este capítulo se describe la implementación del diseño, comentando las diferentes capas de la aplicación, así como elementos no previstos en el diseño original. Su relación con el trabajo global consiste en que este capítulo es aquel en que, partiendo del producto obtenido en la fase de diseño, se implementa la aplicación web aplicando metodologías de desarrollo de software (*Scrum*, *Test Driven Development*).
- Objetivos conseguidos: en este capítulo se analizan los objetivos previstos en la planificación inicial, de la PEC 1, y el objetivo global del TFG, determinando qué objetivos se han alcanzado y cuáles no. Su relación con el trabajo global radica en evaluar los objetivos propuestos, y su consecución.
- Trabajo futuro y posibles mejoras: este capítulo describe qué características se podrían añadir al proyecto como mejora, tanto en el aspecto funcional como no funcional (automatización de pruebas, evaluación de la calidad del código, seguridad,…). Su relación con el trabajo a nivel global radica en analizar cómo se podría mejorar el diseño del producto, tanto internamente en el código o en la seguridad como de cara al cliente.

- Conclusiones: este capítulo describe las reflexiones del autor del TFG a partir del trabajo realizado a lo largo del semestre, siendo su relación global con el trabajo analizar las ideas obtenidas tras desarrollar el TFG, enmarcándolo en el contexto de los estudios llevados a cabo en el Grado en Ingeniería Informática.

2. Análisis

2.1 Análisis de necesidades del proyecto

El proyecto consiste en desarrollar una aplicación que permita a los negocios de las librerías gestionar su trabajo diario, y permitir a los clientes interaccionar vía web con la librería.

La aplicación por un lado permite a los empleados de una librería gestionar los libros que vende, encargar libros a los almacenes, realizar, modificar y anular reservas de libros para clientes, gestionar los foros de la aplicación y participar en ellos, entre otras funcionalidades.

Por otro lado, a los clientes (lectores) la aplicación les permite registrarse en la misma; realizar, modificar y anular reservas de libros; participar en foros, crear foros y participar en ellos, con el objetivo de lograr comunicación entre los clientes y también con los empleados, intercambiando experiencias relacionadas con el mundo de la lectura; entre otras funcionalidades.

Respecto a la reserva de libros por parte de los clientes, no se permite el envío a domicilio con el objetivo de conocer a los clientes y de que los clientes conozcan la librería y la visiten.

2.2 Análisis de requisitos

2.2.1 Requisitos no funcionales

En primer lugar se describen los requisitos no funcionales de la aplicación, que representan características intangibles del sistema:

- NF-1: Se debe permitir el acceso a usuarios de forma concurrente hasta 100 usuarios.
- NF-2: los usuarios de la aplicación web estarán familiarizados con el uso de las tecnologías web a nivel de usuario.
- NF-3: la interfaz de la aplicación debe resultar intuitiva y amigable para el usuario.
- NF-4: la aplicación se desplegará en un servidor de aplicaciones Wildfly, alojado en un servidor físico con las siguientes características hardware y software: 16 GB RAM, procesador i7, sistema operativo Ubuntu Server o Redhat.
- NF-5: la base de datos de la aplicación se alojará en MySQL, configurado en un servidor físico con las siguientes características hardware y software: 16 GB RAM, procesador i7, sistema operativo Ubuntu Server o Redhat.

- NF-6: en el servidor que aloja el servidor de aplicaciones se implantará como servidor web Apache, con la implantación de un certificado que permita conexión segura (https) a la aplicación web.
- NF-7: las contraseñas de los usuarios se almacenarán cifradas en base de datos, y nunca se enviarán en claro cuando se envíen entre las diferentes capas de la aplicación
- NF-8: el acceso a la aplicación se controlará mediante usuario (email) y contraseña, que debe cumplir los requisitos de complejidad: 8 caracteres, una mayúscula, una minúscula, un número y un carácter especial.

2.2.2 Requisitos funcionales

Los requisitos funcionales obtenidos tras reuniones con el cliente se documentan mediante las funcionalidades que debe cubrir la aplicación, los casos de uso que describen las funcionalidades y los diagramas de interacción.

2.2.2.1 Funcionalidades de la aplicación

A continuación se documenta las funcionalidades que debe cubrir la aplicación, organizadas por perfiles:

- Administrador
    - Gestión de usuario: login, logout, actualizar datos del perfil (email, nombre,…), cambio de contraseña.
    - Mantenimiento tablas
        - IVA: gestionar el / los impuesto/s aplicado/s.
        - Crear usuarios de los diferentes perfiles: Administrador, GestorLibros, Vendedor, Lector.
        - Reactivar usuarios dados de baja (de todos los perfiles).
        - Consultar el listado de usuarios existentes filtrando por perfil.
- GestorLibros
    - Gestión de usuario: login, logout, actualizar datos del perfil (email, nombre,…), cambio de contraseña.
    - Mantenimiento tablas
        - Crear usuarios de los diferentes perfiles: GestorLibros, Vendedor, Lector.
        - Reactivar usuarios dados de baja (de todos los perfiles excepto Administrador).

        - Consultar el listado de usuarios existentes filtrando por perfil.
        - IVA: gestionar el / los impuesto/s aplicado/s.
    - Módulo gestor
        - Dar de alta y de baja almacenes.
        - Consultar el listado de almacenes existentes, permitiendo acceder a su edición, tanto si el almacén está de alta o de baja.
        - Dar de alta y de baja libros, así como su edición.
        - Consultar el listado de libros existentes, tanto activos como de baja, permitiendo acceder a su edición.
        - Dar de alta y de baja clientes (lectores), así como editar la ficha de lectores.
        - Gestión de mensajes: permite a los usuarios de este perfil crear mensajes destinados a uno o más clientes concretos, o a todos los clientes activos (no dados de baja), para avisar de información importante, como libros que puedan ser de interés, noticias del mundo de la lectura (publicación de libros, revistas), etc.
    - Módulo de vendedor
        - Reservar libros para un cliente, permitiendo anular una reserva o modificarla.
        - Encargar libros al almacén.
        - Consultar los libros recibidos del almacén y los pendientes de recibir.
        - Registrar la recepción de libros por parte del almacén.
        - Consultar el listado de libros existentes en la aplicación, con filtros por título, autor, ISBN, fecha de publicación, … No debe permitir la edición ni dar de baja o de alta libros. Debe permitir la impresión de la ficha de un libro y del listado de libros, así como la exportación a Excel de listado.
        - Consultar la ficha de clientes, filtrando por DNI, nombre o email, permitiendo consultar las reservas de libros asociadas a cada cliente.

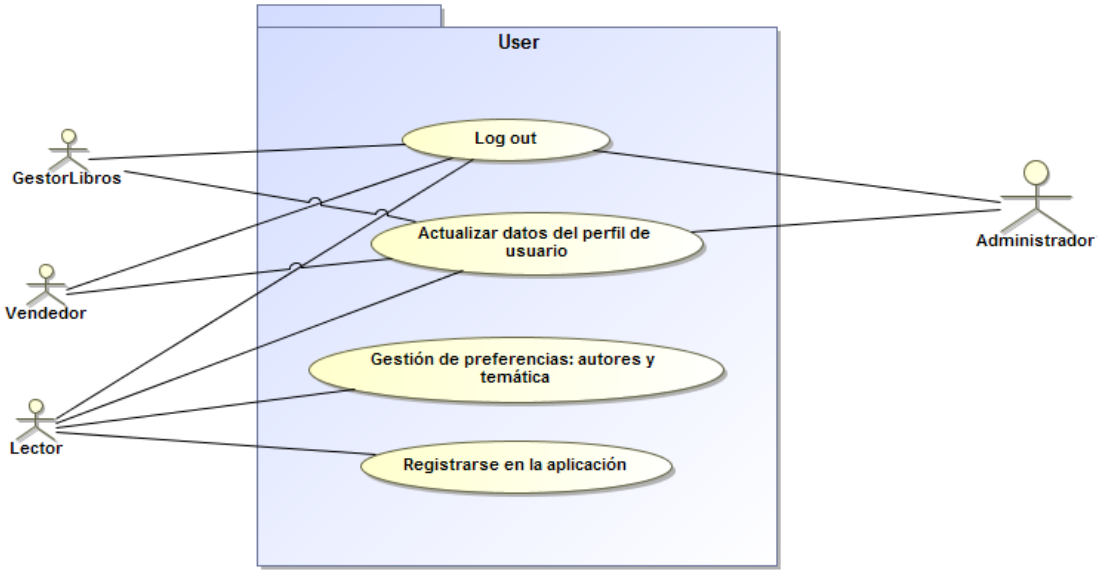
        - Generar una factura (debe permitir añadir varios conceptos -uno por libro comprado-, calcular el total de la factura y aplicar el IVA correspondiente), y por último asociarla a un cliente.
        - Eliminar una factura.
        - Imprimir una factura.
        - Listar las facturas, con filtros por cliente y fecha de la factura.
    - Módulo de comunicación
        - Dar de alta y de baja foros.
        - Enviar mensajes al foro.
        - Eliminar mensajes.
        - Modificar el perfil de un usuario para evitar el envío de mensajes al foro, por ejemplo, se puede dar esta situación si un usuario (lector) envía mensajes que no son apropiados. En un paso posterior, se puede considerar la baja del usuario en la aplicación.

- Vendedor
    - Gestión de usuario: login, logout, actualizar datos del perfil (email, nombre,…), cambio de contraseña.
    - Módulo de vendedor
        - Reservar libros para un cliente, permitiendo anular una reserva o modificarla.
        - Encargar libros al almacén.
        - Consultar los libros recibidos del almacén y los pendientes de recibir.
        - Registrar la recepción de libros por parte del almacén.
        - Consultar el listado de libros existentes en la aplicación, con filtros por título, autor, ISBN, fecha de publicación, … No debe permitir la edición ni dar de baja o de alta libros. Permitir la impresión de la ficha de un libro y del listado de libros.
        - Consultar la ficha de clientes, filtrando por DNI, nombre o email, permitiendo consultar las reservas de libros asociadas a cada cliente.

        - Generar una factura (debe permitir añadir varios conceptos -uno por libro comprado-, calcular el total de la factura y aplicar el IVA correspondiente) y asociarla a un cliente.
        - Eliminar una factura.
        - Imprimir una factura.
        - Listar las facturas, con filtros por cliente y fecha de la factura.
    - Módulo de comunicación
        - Dar de alta y de baja foros.
        - Enviar mensajes al foro.
- Lector (cliente)
    - Gestión de usuario / cliente: login, logout, registrarse, cambio de contraseña, editar y consultar datos del cliente (NIF, dirección, teléfono, email,…). Consultar libros comprados y la factura. También debe permitir consultar facturas por fecha, permitiendo la impresión de la misma.
    - Gestión de reservas: Realizar una reserva, Anular reserva, Encargar un libro (no hay en stock), Modificar y/o anular la solicitud de un libro (con un máximo de 48 horas de antelación a la fecha de recepción en la tienda).
    - Gestión de libros: Consultar libros existentes en la aplicación, filtrando por Título, Autor, Fecha de publicación, ISBN,… así como la consulta de la ficha del libro. Debe permitir la impresión y exportación a formato Excel del listado de libros buscados, así como de la ficha.
    - Comunicación: debe permitir al usuario (lector) consultar los foros existentes, crear foros y enviar mensajes.
    - Bandeja de mensajes: en esta funcionalidad el cliente recibe información relevante, como puede ser: anulación y/o modificación de reservas y/o de solicitudes de libro; avisos de información relevante: publicación de libros, de revistas; avisos relevantes para el mundo de la lectura; y cualquier aviso que desee comunicar el personal de la librería.

2.2.2.2 Casos de uso de gestión de usuario

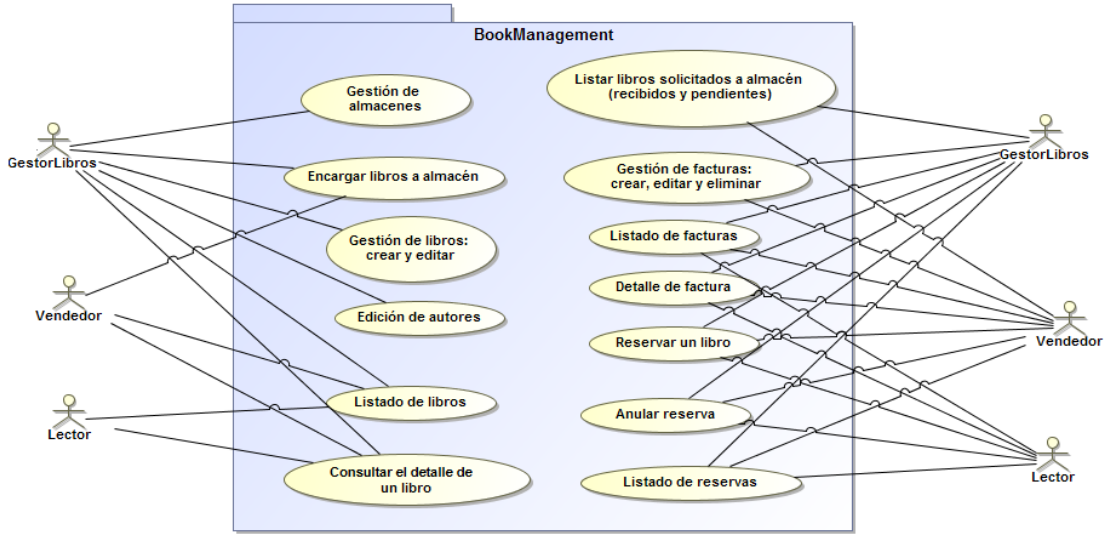
En esta sección se describen gráficamente los casos de uso relacionados con la gestión de perfiles de usuario: actualización de datos personales, actualización de preferencias y cambio de contraseña.



No se documentan textualmente puesto que se acuerda con el cliente que son funcionalidades comunes en muchas aplicaciones. Se recuerda únicamente que la gestión de preferencias debe permitir configurar a un usuario qué autores y libros le gustan.

2.2.2.3 Casos de uso de gestión de libros

En esta sección se describen gráficamente y textualmente los casos de uso relacionados con la gestión de libros.



Caso de uso: Gestión de almacenes
Actor principal: Usuario de perfil GestorLibros (todos los pasos), Usuario de perfil Vendedor (pasos 1, 2 y 3)
Ámbito: Sistema informático de gestión de librerías (GLIB)

Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas: -
Garantías en caso de éxito (postcondiciones): El sistema mostrará los almacenes existentes, permitiendo la creación, edición y/o eliminación de almacenes
Escenario principal de éxito:

1) El usuario accede al módulo Gestor de libros
2) El usuario accede a la opción de menú “Gestión de almacenes / Listado de almacenes”
3) El sistema muestra un formulario de búsqueda, con un seleccionable que muestra el nombre de cada almacén.
4) El usuario elije un almacén, o la opción Todos, y el sistema muestra el listado de almacenes, mostrando Nombre, dirección y página web.
5) El usuario (perfil GestorLibros) crea un nuevo almacén
6) El sistema muestra el formulario de alta de un almacén
7) El usuario (perfil GestorLibros) debe introducir los datos del nuevo almacén: Nombre, CIF, Dirección, página web, teléfono de contacto, persona de contacto, si está activo (por defecto lo está) y fecha de alta (por defecto el día actual). A continuación, confirma el alta.
8) El sistema informa del alta y muestra el listado actualizado de almacenes al usuario.

Escenarios alternativos:

3a) El usuario accede al listado de libros solicitados al almacén (Gestión de almacenes / Listado libros solicitados).

3a1) El sistema muestra el formulario para buscar los libros solicitados al almacén, indicando el almacén y el estado, que puede ser pendiente o recibido.

3a2) El usuario busca datos eligiendo como estado de los libros solicitados: pendiente o recibido

3a2a) Si sucede algún error, el sistema informa al usuario, permitiendo buscar datos de nuevo

3a3) El sistema muestra el listado de los libros según los filtros elegidos por el usuario. De cada libro se muestra: Almacén (suministrador), Título, Autor, ISBN y precio.

3b) El usuario accede al detalle del almacén

3b1) El sistema muestra el detalle del almacén, visualizando en pantalla todos los datos disponibles del almacén (ver paso 7 del escenario principal de éxito para más información).

3b2) El usuario edita el almacén.

3b3) El sistema muestra un formulario con los datos actuales del almacén, permitiendo editarlos.

3b4) El usuario guarda los cambios.

3b5) El sistema confirma el correcto guardado.

6a) El usuario (perfil GestorLibros) cancela el proceso de alta de un nuevo almacén.

6a1) El sistema informa de la cancelación del proceso y regresa al listado de almacenes

6b) El usuario (perfil GestorLibros) elimina un almacén.

6b1) El sistema solicita confirmación al usuario de la eliminación
6b2) El usuario confirma la eliminación
6b2) El sistema informa al usuario de la eliminación, mediante borrado lógico, y regresa al listado de almacenes para mostrárselo actualizado al usuario

Caso de uso: Encargar libros a almacén
Actor principal: Usuarios de perfil GestorLibros y de perfil Vendedor
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas: -
Garantías en caso de éxito (postcondiciones): El sistema permitirá al usuario encargar libros a almacén
Escenario principal de éxito:

1) El usuario accede al módulo Gestor de libros.
2) El usuario accede a la opción de menú “Gestión de almacenes / Encargo de libros”.
3) El sistema muestra una tabla donde se debe elegir o introducir: libro y cantidad. En la tabla se permite añadir filas para crear más de un encargo.
4) El usuario confirma el encargo.
5) El sistema solicita confirmación al usuario.
6) El usuario confirma el encargo.
7) El usuario accede a la opción del menú “Gestión de almacenes / Listado de libros solicitados”.
8) El sistema muestra un formulario donde se debe elegir el almacén y el estado de la entrega (Pendiente, Recibido).
9) El usuario elije el almacén y el estado, y el sistema muestra los encargos correspondientes.
10) En el caso de buscar encargos pendientes, el sistema permite registrar la entrega al usuario.
11) El usuario registra la entrega.
12) El sistema solicita confirmación.
13) El usuario confirma el registro.
14) El sistema actualiza el estado del encargo, y el stock de los libros encargados (aumentado la cantidad teniendo en cuenta la encargada).

Escenarios alternativos:
7a) El usuario cancela el encargo, en cuyo caso el sistema permanece en la pantalla del paso 4.
14a) El usuario cancela el registro, en cuyo caso el sistema permanece en el paso 11.

Caso de uso: Gestión de libros
Actor principal: Usuario de perfil GestorLibros (todos los pasos), Vendedor y Lector (pasos 1, 2 y 3 excepto 3b)

Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas: --
Garantías en caso de éxito (postcondiciones): El sistema mostrará al usuario los libros existentes, permitiendo la creación y/o edición de libros.
Escenario principal de éxito:

1) El usuario accede al módulo Gestor de libros.
2) El usuario accede a la opción Gestión de libros.
3) El sistema muestra el listado de libros, con los filtros de búsqueda autor y libro, mostrando los datos: ISBN, Título, Autor, Editorial, Precio y Stock.
4) El usuario, de perfil GestorLibros, accede a la creación de un nuevo libro (Gestión de libros / Alta de libro).
5) El sistema muestra el formulario de alta de un nuevo libro, donde el usuario debe introducir: ISBN, Título, Año, Páginas, Editorial, temática y precio. Se debe incluir también un pequeño resumen del libro. Así mismo, se debe introducir la siguiente información del autor: Nombre, Apellidos, biografía y fecha de nacimiento. También se indica si el libro está descatalogado o no, al dar de alta un libro este campo tendrá por defecto el valor “No”, pero de esta forma se permite descatalogar libros. Al crear un nuevo libro, el valor por defecto del stock es 0 y no está descatalogado.
6) El usuario acepta el alta y el sistema solicita confirmación al usuario.
7) El usuario confirma y el sistema informa del correcto proceso de alta del libro.

Escenarios alternativos:

3a) El usuario accede al detalle del libro

3a1) El sistema muestra el detalle con la siguiente información: ISBN, Título, Autor, Año, Páginas, Editorial, Edición, Precio y Stock. También se muestra un pequeño resumen y una pequeña biografía del autor. Se permite al usuario (de perfil GestorLibros) acceder a la edición.

3b) El usuario, de perfil GestorLibros, edita el libro.

3b1) El sistema muestra el detalle con la siguiente información: ISBN, Título, Autor, Año, Páginas, Editorial, Edición Temática, Precio y Stock. También se muestra un pequeño resumen, el nombre y apellidos del autor, y una pequeña biografía del mismo. En este caso se permite al usuario la edición de los datos del libro mostrados, pero no del autor.

3b1a) El usuario guarda los cambios.

3b1a1) El usuario cancela la edición

3b1a2) El sistema regresa al detalle del libro

3b1b) El sistema informa del correcto guardado y muestra el detalle actualizado del libro.

6a) El usuario cancela el proceso de creación / alta

6a1) El sistema solicita confirmación de la cancelación al usuario

6a2) El sistema informa de la cancelación y regresa al listado de libros (paso 3)

Caso de uso: Edición de autores

Actor principal: Usuario de perfil GestorLibros
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema permite editar autores existentes
Escenario principal de éxito:

1) El usuario accede al módulo Gestor de libros
2) El usuario accede a Gestión de libros / Edición de autores
3) El sistema muestra un formulario con un seleccionable, para elegir un autor.
4) El usuario elije un autor.
5) El sistema muestra los datos del autor (nombre, apellidos, biografía y fecha de nacimiento), permitiendo su edición.
6) El usuario modifica los datos que desea y guarda los cambios.
7) El sistema solicita confirmación del guardado al usuario.
8) El usuario confirma los cambios.
9) El sistema informa del correcto guardado y sale de la funcionalidad.

Escenarios alternativos:

8a) El usuario cancela el proceso de edición de un autor.

8a1) El sistema informa de la cancelación del proceso y abandona la pantalla de edición.

Caso de uso: Gestión de reservas
Actor principal: Usuarios de perfil: GestorLibros, Vendedor, Lector
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema reserva una cantidad N de uno o más libros para un cliente
Escenario principal de éxito:

1) El usuario accede al módulo Gestión de libros
2) El usuario accede a la opción Gestión de reservas
3) El sistema muestra el listado de reservas, permitiendo filtrar por cliente (lector), título y autor, estado (Pendiente, Sin stock, Disponible, Anulado o Entregado) y fecha de reserva. Se muestra la siguiente información de cada reserva: cliente (lector), título y autor, cantidad reservada y fecha de reserva, ordenado por cliente y fecha de reserva por defecto.
4) El usuario crea una nueva reserva (Gestión de reservas / Reservar un libro)
5) El sistema muestra el formulario de reserva de libros. El usuario debe introducir: cliente, título del libro, y la cantidad a reservar.

6) El usuario confirma la reserva y el sistema informa de la correcta reserva. Se debe ajustar el stock del libro reservado en función de la cantidad reservada.
7) El usuario anula una reserva.
8) El sistema muestra el formulario para buscar reservas susceptibles de ser anuladas. El usuario debe introducir: cliente, título y autor del libro.
9) El sistema muestra las reservas activas (pendientes de entrega) para los datos introducidos en el paso 8.
10) El usuario anula la reserva, regresando la cantidad de la reserva al stock.
11) El sistema informa de la anulación.

Escenarios alternativos:

3a) Si el usuario es de los perfiles Vendedor o GestorLibros, permitirá elegir al cliente. En caso contrario, el cliente vendrá pre-seleccionado por defecto (perfil Lector).

5a) Si el usuario es de los perfiles Vendedor o GestorLibros, permitirá elegir al cliente. En caso contrario, el cliente vendrá pre-seleccionado por defecto (perfil Lector).

5a1) Si no hay stock, se informará el usuario y se actualizará el estado de la reserva para indicar que no hay stock, en cuyo caso se enviará un mensaje personal al usuario o a los usuarios (de perfil GestorLibros o Vendedor) que se desee para que gestionen la solicitud a almacén del libro, y actualicen el estado de la reserva. Se debe informar al usuario del tiempo previsto hasta la recepción del libro.

6a) Si se produce algún error, el sistema informa al usuario y regresa al formulario de reserva

6b) El usuario cancela la reserva.

6b1) El sistema informa de la cancelación del proceso y regresa al paso 3.

9a) Si el usuario es de los perfiles Vendedor o GestorLibros, permitirá elegir al cliente. En caso contrario, el cliente vendrá pre-seleccionado por defecto (perfil Lector).

10a) El usuario cancela el proceso y el sistema regresa al paso 8.

Caso de uso: Gestión de facturas
Actor principal: usuarios de perfil: GestorLibros, Vendedor, Lector
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema muestra el listado de facturas, permitiendo la creación, edición y eliminación
Escenario principal de éxito:
1) El usuario accede al módulo Gestor de libros (perfiles Vendedor y GestorLibros).
2) El usuario accede a la opción Gestión de facturas, si es Vendedor o Gestor Libros. Si es cliente la opción se denomina Listado de facturas.

3) El sistema muestra un formulario para buscar facturas, con los filtros: cliente (DNI o nombre y apellidos), fecha de factura, número de empleado que generó la factura.
4) El usuario elije los filtros del formulario del paso 3 y busca datos.
5) El sistema muestra los datos encontrados, indicando: Cliente de la factura (DNI), Número de empleado que la generó y fecha de la factura. Si el número de datos es elevado se debe mostrar paginado.
6) El usuario accede al detalle de la factura.
7) El sistema muestra la siguiente información de la factura: Cliente de la factura (NIF), Número de empleado que la generó y fecha de la factura, así como cada concepto de la factura (título y autor del libro comprado), cantidad, importe individual (en función de la cantidad de cada libro) y el importe total.
8) El usuario, de perfiles GestorLibros o Vendedor, crea una nueva factura (Gestión de facturas / Crear factura).
9) El sistema muestra el formulario de alta de facturas, donde el usuario debe introducir: Número de empleado (se determina a partir del usuario que ha iniciado sesión, ya que es igual al NIF), fecha de la factura (por defecto el día actual) y generar un concepto por cada libro comprado, indicando la cantidad de cada libro. El usuario debe introducir el cliente de la factura (NIF), eligiéndolo de una lista si es un cliente registrado, o introduciendo el NIF en un cuadro de texto. En el caso de no ser un cliente registrado, al crear la factura no se guardará información del cliente.
10) El usuario confirma la creación y el sistema permite imprimir la factura.
11) El usuario, de perfiles GestorLibros o Vendedor, elimina una factura (desde el listado de facturas).
12) El sistema solicita confirmación al usuario y elimina la factura, mediante borrado lógico para mantener un historial de facturas, e informa al usuario de la eliminación, regresando al paso 3.

Escenarios alternativos:

5a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

7a) El usuario, de perfiles GestorLibros o Vendedor, accede a la edición de la factura, mostrándose un formulario con todos los datos de la factura, pudiendo modificarlos, así como eliminar o añadir conceptos en la misma.

7a1) El usuario modifica datos y guarda los cambios.

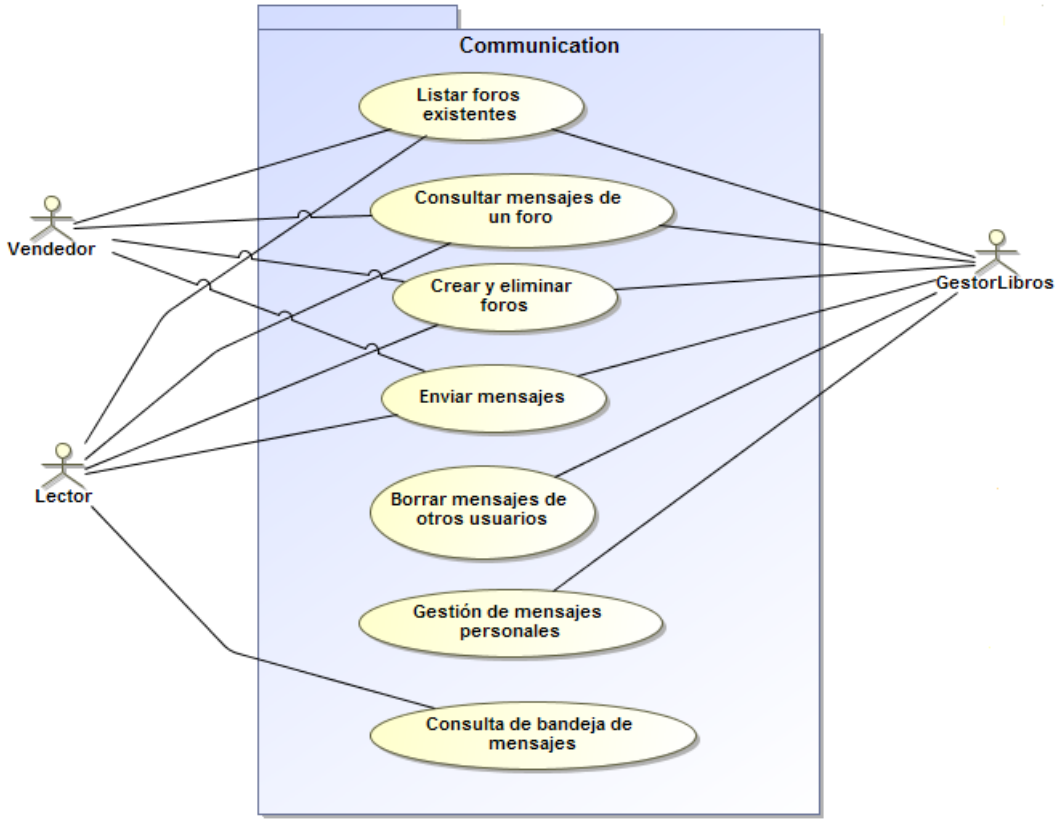
7a2) El sistema informa del correcto guardado y regresa al detalle de la factura actualizado.

10a) El usuario cancela la creación de la factura.

10a1) El sistema regresa al paso 3.

2.2.2.4 Casos de uso de comunicación

En esta sección se describen gráfica y textualmente los casos de uso relacionados con la comunicación de los usuarios de la aplicación.



Caso de uso: Gestión de foros
Actor principal: Usuarios de perfil: GestorLibros, Vendedor o Lector
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema permite consultar y/o crear foros a usuarios de perfil Gestor Libros, Vendedor o Lector, así como el envío de mensajes. A usuarios de perfil GestorLibros, se les permite eliminar mensajes enviados a un foro si no son adecuados.
Escenario principal de éxito:

1) El usuario accede al módulo Comunicación, en el caso de los usuarios de perfil Vendedor y GestorLibros.
2) El usuario accede a la opción Gestión de foros.
3) El sistema permite buscar al usuario datos, filtrando por temática del foro, y se muestra la siguiente información: Nombre del foro, usuario que lo creó y fecha de creación.
4) El usuario da de alta un nuevo foro, introduciendo nombre del foro y temática.
5) El usuario accede a un foro.
6) El sistema muestra los mensajes enviados en dicho foro, ordenados por fecha de creación, con la siguiente información: Nombre del usuario autor del mensaje, contenido del mensaje y fecha de envío del mensaje. Se permite acceder al envío de mensajes.
7) El usuario envía un mensaje.

8) El sistema muestra el formulario de envío de mensajes, permitiendo introducir el contenido del mensaje.
9) El usuario confirma el envío del mensaje.
10) El sistema muestra al usuario el foro con los mensajes actualizados.

Escenarios alternativos:

4a) El usuario elimina un foro.

4a1) El sistema comprueba si el foro tiene mensajes asociados.

4a1a) Si tiene mensajes asociados, el sistema informa al usuario que no se puede eliminar el foro.

4a1b) Si no tiene mensajes asociados, el sistema solicita confirmación al usuario y elimina el foro (borrado lógico).

6b) Si el usuario es de perfil GestorLibros, borra un mensaje (si el contenido no es apropiado o por otros motivos).

6b1a) El sistema solicita confirmación al usuario y se realiza un borrado lógico del mensaje, para que quede como histórico.

6b1a) El usuario cancela el borrado, el sistema regresa al paso 3 (listado de foros).

Caso de uso: Gestión de mensajes personales
Actor principal: Usuarios de perfil GestorLibros, Lector
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema permite al usuario de perfil GestorLibros crear un mensaje nuevo y al usuario de perfil Lector consultar los mensajes recibidos.
Escenario principal de éxito:

1) El usuario accede al módulo Comunicación.
2) El usuario de perfil GestorLibros accede a Gestión de mensajes personales
3) El sistema muestra un formulario para buscar mensajes personales con los filtros: NIF de usuario y fecha de envío.
4) El usuario introduce un NIF y/o una fecha de envío y busca datos.
5) El sistema muestra los resultados obtenidos, con paginación si superan un número de resultados prefijado. Los datos mostrados son: usuario al que se envió, asunto, contenido del mensaje y fecha de envío.
6) El usuario, de perfil GestorLibros, crea un nuevo mensaje.
7) El sistema muestra un formulario de creación de mensajes. El usuario debe introducir: NIF del destinatario, asunto y contenido del mensaje.
8) El usuario introduce los datos y confirma la creación del mensaje.
9) El sistema informa del correcto envío del mensaje y regresa a la pantalla de inicio del perfil GestorLibros.
10) El usuario, de perfil Lector, accede al módulo Comunicación, a la funcionalidad Bandeja de mensajes.
11) El sistema muestra los mensajes recibidos, destacando aquellos no leídos, permitiendo filtrar por un rango de fechas. Si el número de

mensajes es elevado, se debe permitir la paginación. La información mostrada por cada mensaje es: Fecha de envío y asunto del mensaje.

12)El usuario accede al detalle de un mensaje. El sistema muestra toda la información del mensaje: Usuario que lo envió, fecha de envío, asunto y contenido del mensaje.

Escenarios alternativos:

5a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

6a) El usuario cancela el proceso de alta de un nuevo mensaje.

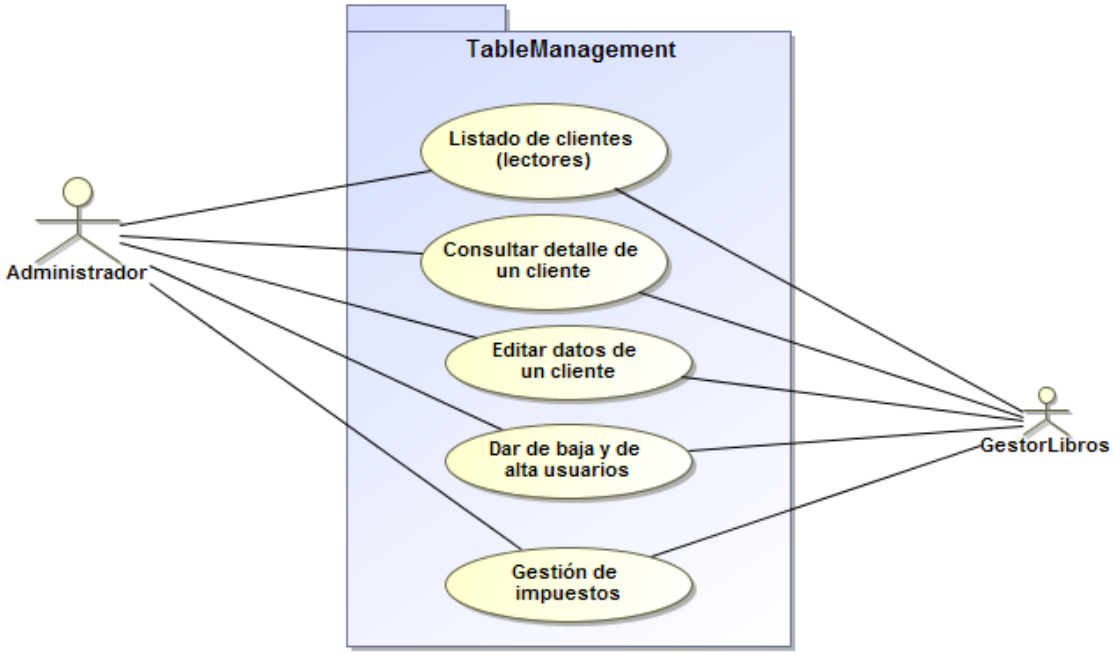
6a1) El sistema informa de la cancelación del proceso y regresa al listado de mensajes.

9a) El sistema informa al usuario si no se envía correctamente el mensaje.

11a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

2.2.2.5 Casos de uso de gestión de tablas

En esta sección se describen gráficamente y textualmente los casos de uso relacionados con la gestión de tablas.



Caso de uso: Gestión de clientes
Actor principal: Usuarios de perfil GestorLibros o Administrador
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema muestra el listado de clientes al usuario, en función de los filtros introducidos, y permite tanto acceder al detalle como a la edición de los datos del cliente.
Escenario principal de éxito:

1) El usuario accede al módulo Gestión de tablas.
2) El usuario accede a la funcionalidad Gestión de clientes.
3) El sistema muestra un formulario para buscar clientes, con los filtros NIF (incluyendo la opción Todos), fecha de alta, o Email.
4) El usuario introduce los filtros de búsqueda que desea y busca datos.
5) El sistema muestra los resultados, paginando si el número de resultados es elevado. Los datos que muestra de cliente son: NIF, Email y Nombre y Apellidos.
6) El usuario consulta el detalle de un cliente.
7) El sistema muestra todos los datos del cliente: NIF, Nombre y Apellidos, Email, Teléfonos de contacto, Dirección y sus preferencias de lectura (autores y temáticas).
8) El usuario accede a la edición del cliente.
9) El sistema permite editar los datos del cliente mostrados en el paso 7, excepto las preferencias de lectura. Adicionalmente, es posible bloquear a un usuario para que no pueda enviar mensajes en los foros.
10) El usuario modifica los datos que desea y acepta los cambios.
11) El sistema solicita confirmación al usuario de los cambios y el usuario confirma.
12) El sistema informa del correcto guardado del cliente y regresa a la pantalla del paso 3.

Escenarios alternativos:

5a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

7a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

10a) El usuario cancela el proceso de edición de un cliente.

10a1) El sistema informa de la cancelación del proceso y regresa a la pantalla del paso 3.

12a) El sistema informa al usuario si no se guarda correctamente el cliente.

Caso de uso: Alta y baja de usuarios
Actor principal: Usuario de perfil GestorLibros o Administrador
Ámbito: Sistema informático de gestión de librerías (GLIB)
Nivel de objetivo: Usuario
Stakeholders e intereses:
Precondición: El usuario debe haber iniciado sesión
Garantías mínimas:
Garantías en caso de éxito (postcondiciones): El sistema permite el correcto alta o baja de usuarios de los diferentes perfiles de la aplicación.
Escenario principal de éxito:

10) El usuario accede al módulo Gestión de tablas.
11) El usuario accede a la opción Alta y baja de usuarios.
12) El sistema muestra una pantalla que permite buscar usuarios en función del perfil: GestorLibros, Vendedor, Administrador o Lector.
13) El usuario elije un perfil y busca datos.

14) El sistema muestra los usuarios del perfil elegido, con paginación si el número de resultados es elevado. Se muestra la siguiente información de cada usuario: NIF, Nombre y Apellidos, Email y si está activo o no.
15) El usuario accede al alta de un usuario.
16) El sistema muestra el formulario para dar de alta un usuario, debiendo introducir los siguientes datos: NIF, nombre y apellidos, email, dirección y teléfonos de contacto. Es obligatorio introducir al menos un teléfono de contacto. El resto de campos (NIF, nombre y apellidos, email y dirección) son obligatorios. Se puede indicar si un usuario puede enviar mensajes en los foros. Por defecto, todos los usuarios pueden enviar mensajes en los foros. La contraseña por defecto es igual que el NIF, siendo obligatorio cambiarla en el primer inicio de sesión. Se debe elegir el tipo de perfil.
17) El usuario introduce los datos necesarios y acepta el alta.
18) El sistema guarda el alta del usuario y confirma al usuario el correcto guardado. El sistema asigna por defecto la misma contraseña que el NIF, siendo obligatorio modificarla en el siguiente acceso del usuario. Las contraseñas se deben almacenar cifradas.
19) El usuario da de baja un usuario, marcando como “NO” el campo “Activo” de la información mostrada en el paso 5. A continuación, guarda los cambios.
20) El sistema confirma al usuario los cambios guardados.

Escenarios alternativos:

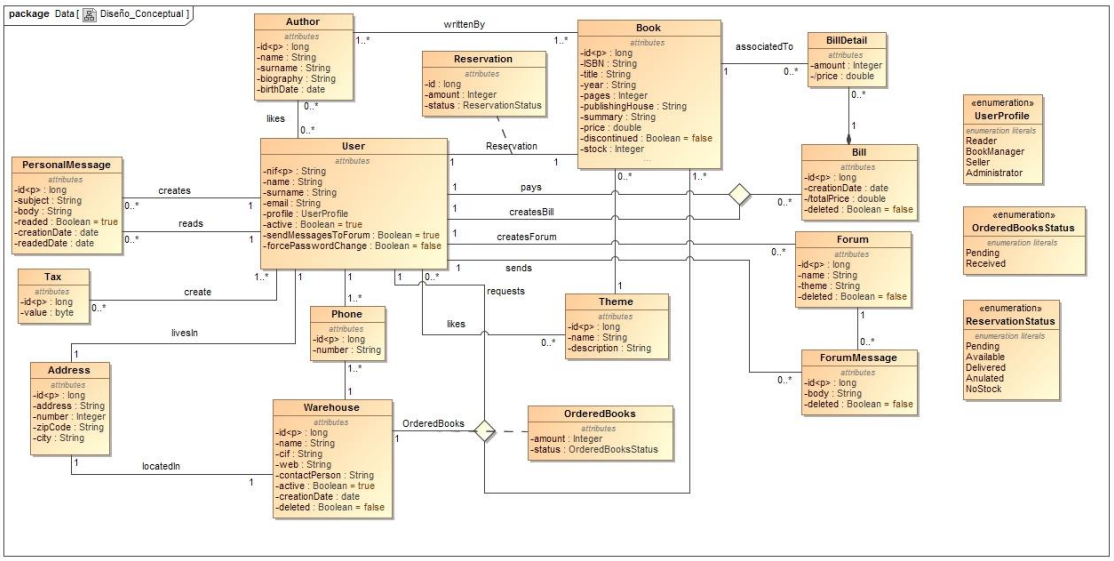
5a) El sistema informa al usuario si no encuentra datos o no es posible obtenerlos.

8a) El usuario cancela el proceso de alta de un nuevo usuario.

8a1) El sistema informa de la cancelación del proceso y regresa al paso 3.

2.3. Diseño conceptual

En este apartado se describe el diseño conceptual del sistema, identificando aquellas entidades del mundo real, con los atributos correspondientes, que se aplican en el dominio de estudio: la librería.



A partir del diseño conceptual, se evalúa si se ha incurrido en alguna trampa de diseño, generando a continuación el diseño relacional de la base de datos. Se comprueba también si el diseño relacional cumple las formas normales, con resultado satisfactorio.

A partir de dicho modelo, se evalúa si se incumple alguna de las trampas de diseño. Se analiza cada trampa de diseño de forma independiente.

En primer lugar se analiza la trampa del abanico, que sucede cuando se tienen tres entidades relacionadas mediante tres tipos de relaciones binarias (1..*), eliminándose el tipo de relación binaria que no se debe. Tras analizar el diseño conceptual, se determina que no se produce esta situación, ya que no se han encontrado casos de esta trampa durante el diseño conceptual, tanto “intencionados” eliminando un tipo de relación binaria como directamente por omisión de un tipo de relación.

En segundo lugar, se analiza la trampa del corte, que sucede en la misma situación que la trampa del abanico, por lo que tampoco se detectan casos en el modelo conceptual.

En tercer lugar, se analiza la trampa de perdida de afiliación, concluyendo que en el modelo conceptual no se han representado relaciones ternarias como tres tipos de relación binaria derivadas de una relación ternaria.

En cuarto lugar, se analiza la trampa de aridad de los tipos de relación, para lo cual se analizan las relaciones ternarias que tengan alguna entidad participante con multiplicidad 1. Las dos relaciones ternarias existentes cumplen esos requisitos:

- La relación ternaria entre User (dos veces) y Bill, que representa la creación de una factura por parte de un Vendedor o un Gestor de libros y el pago por parte de un usuario con perfil Reader (cliente), tiene multiplicidad uno en la entidad User, sin embargo, las entidades participantes no se han incluido por error, ya que un usuario (de perfil Seller o BookManager) crea la factura, otro usuario la paga (perfil Lector

-cliente) y la otra entidad es la factura.
- La relación ternaria entre User, Warehouse y Book representa la solicitud de libros a almacén por parte de un usuario (perfiles Seller o BookManager), por lo que ninguna de las entidades mencionadas se ha incluido de manera errónea.

En quinto y último lugar, se analiza la trampa de "Semántica de los tipos de entidad", para lo cual se revisan los atributos de cada entidad, por si alguno se hubiese incluido de manera errónea, sin detectar ningún caso.

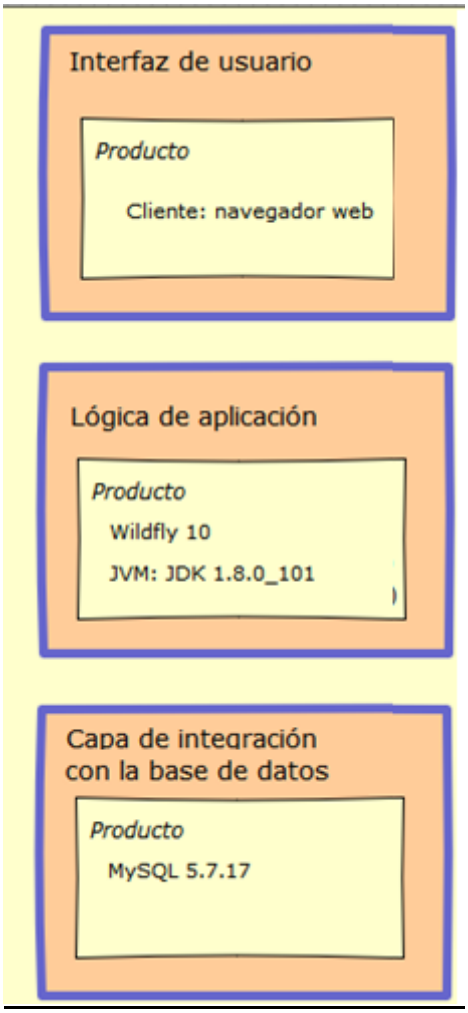
3. Diseño

En esta etapa se documenta el diseño de la arquitectura del sistema, decidiendo que aproximación se implementará para estructura la aplicación web.

3.1. Arquitectura de la aplicación

Se decide escoger como arquitectura principal de la aplicación la arquitectura en tres capas:

- Capa de presentación: Esta capa o nivel se encarga de mostrar al usuario los componentes necesarios que permitan interactuar al mismo con la aplicación, mostrando datos, así como obteniendo y validando los datos introducidos por el usuario. La capa de presentación debe encargarse también de mostrar al usuario únicamente aquellas funcionalidades que tenga asignadas en función de su perfil (Lector, Vendedor, GestorLibros o Administrador). Esta capa se comunica con la de negocio, a la que se notifican las acciones del usuario con la interfaz de la aplicación.
- Capa de negocio: esta capa o nivel representa los procesos del negocio y es la encargada de procesar las acciones del usuario, comunicándose con la capa de administración de datos, cuando sea necesario persistir información en la base de datos.
- Capa de administración de datos o capa de integración: esta capa o nivel se encarga de persistir la información en la base de datos, en respuesta a las acciones del usuario, así como de obtener la información necesaria para la capa de negocio.



3.2. División de la aplicación en componentes

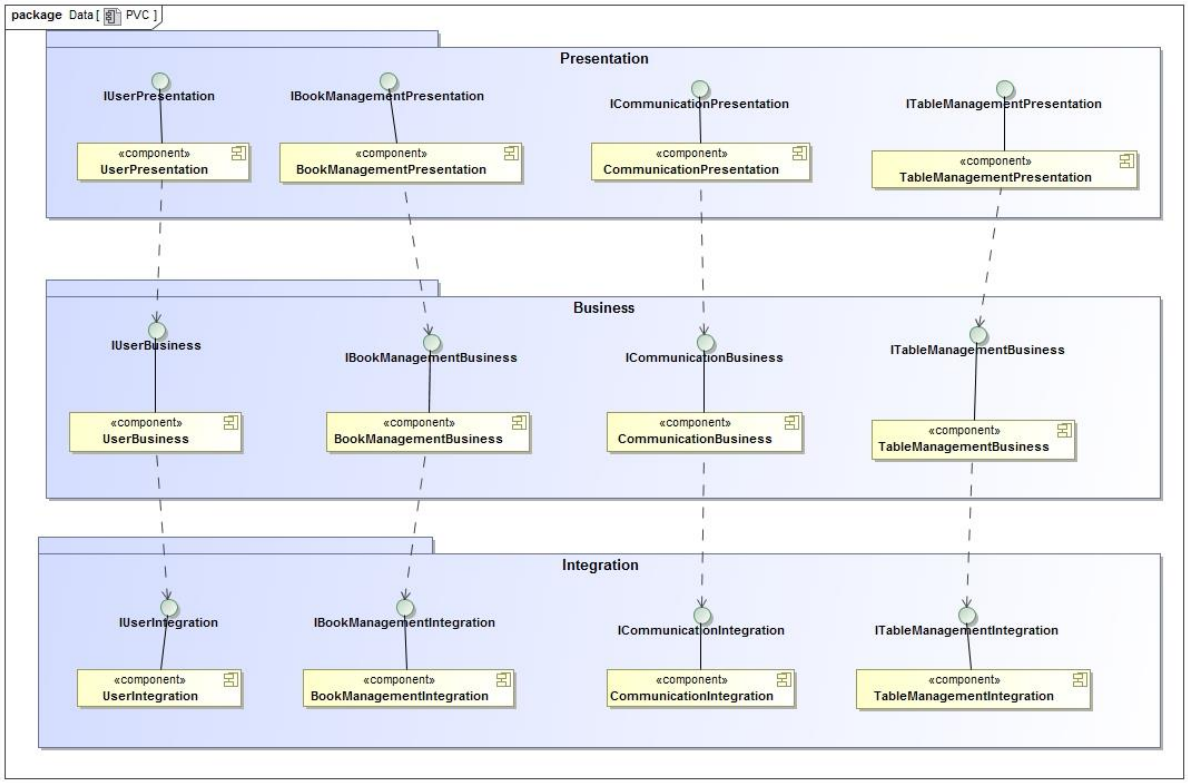
La aplicación se estructura en cuatro componentes (Usuario, Gestión de libros, Comunicación y Gestión de tablas), en relación con los paquetes definidos en el diagrama de casos de uso, y que contienen toda la funcionalidad necesaria de la aplicación. Cada componente se diseña mediante una interfaz y un componente en cada una de las capas en que se divide la aplicación: presentación, negocio e integración.

En la capa de presentación reside la interfaz de la aplicación para interactuar con el usuario (vista) y se encarga de recoger la interacción del usuario con la interfaz (controlador y modelo).

En la capa de negocio reside la lógica de negocio de la aplicación, que trata los datos introducidos por el usuario en la capa de presentación, para transmitirlos a la capa de integración y retornar una respuesta o *feedback* al usuario de la operación realizada.

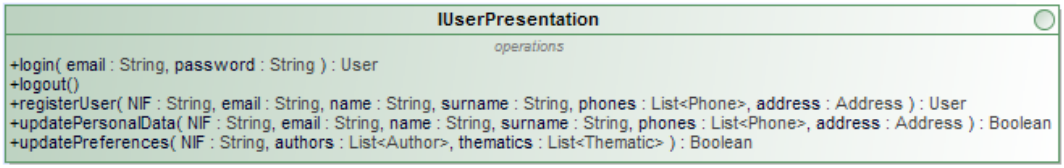
En la capa de integración reside la persistencia de la información que maneja la aplicación.

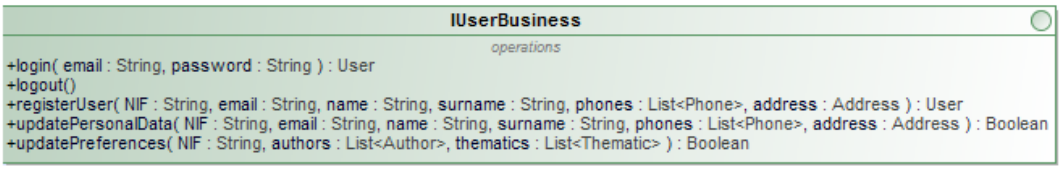
En el diagrama del punto de vista de la computación, no se incluye el detalle de cada interfaz para facilitar la legibilidad del diagrama.

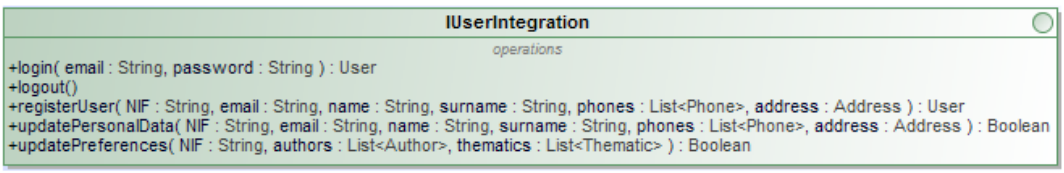


Punto de vista de la computación

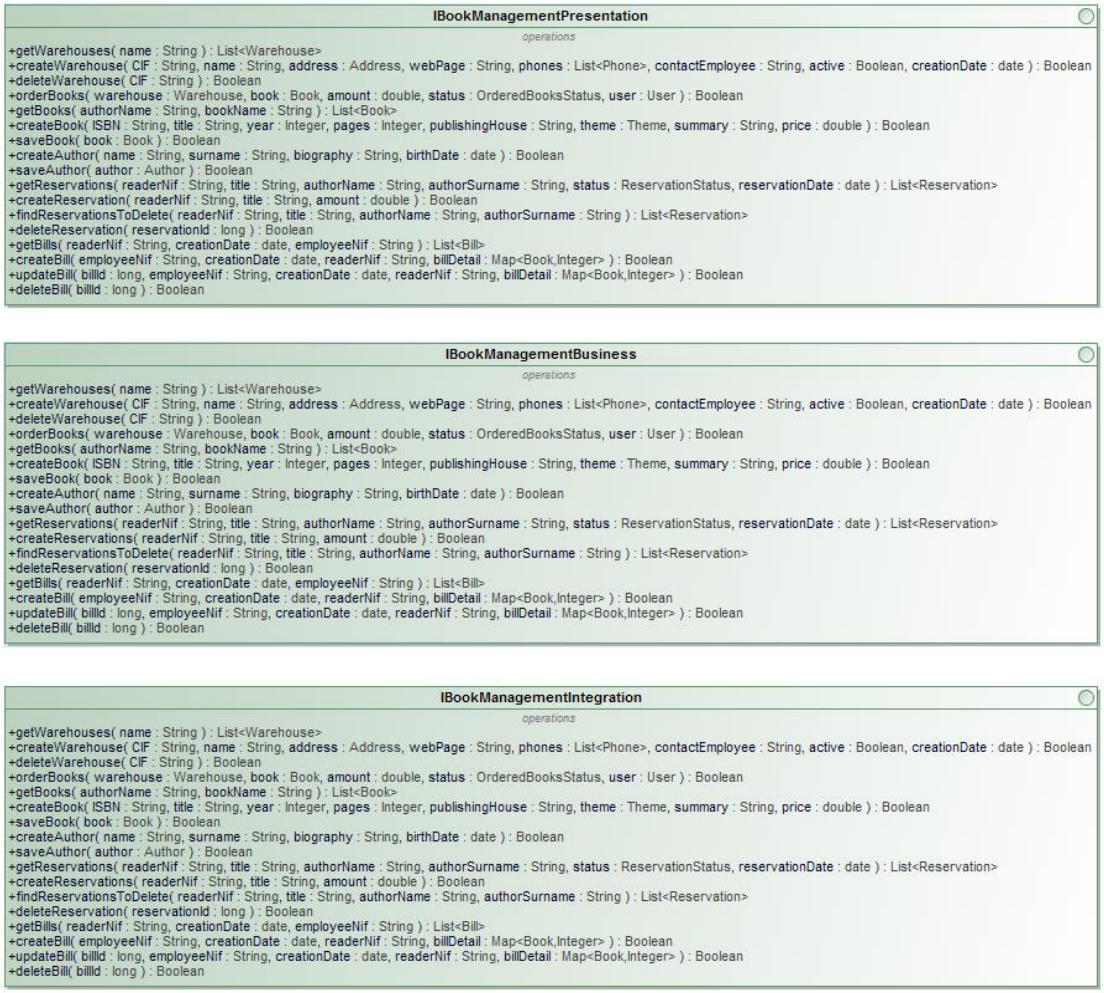
Se muestran a continuación en detalle las interfaces de cada componente por capa en que se estructura la aplicación.



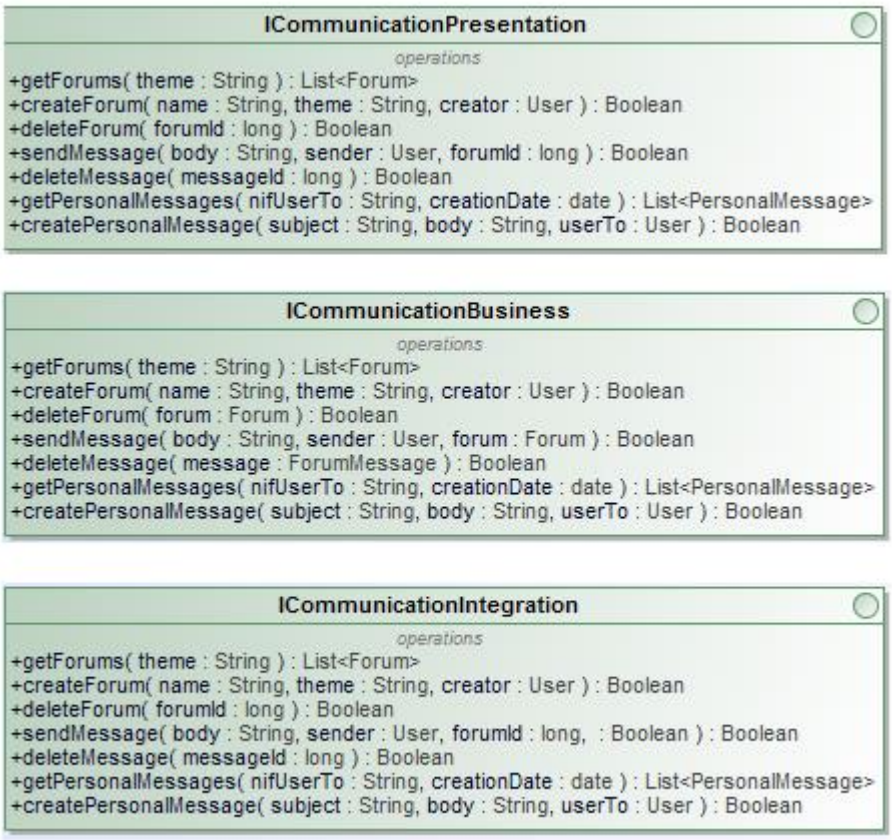




Detalle de interfaces en el componente User



Detalle de interfaces en el componente BookManagement



Detalle de interfaces en el componente Communication

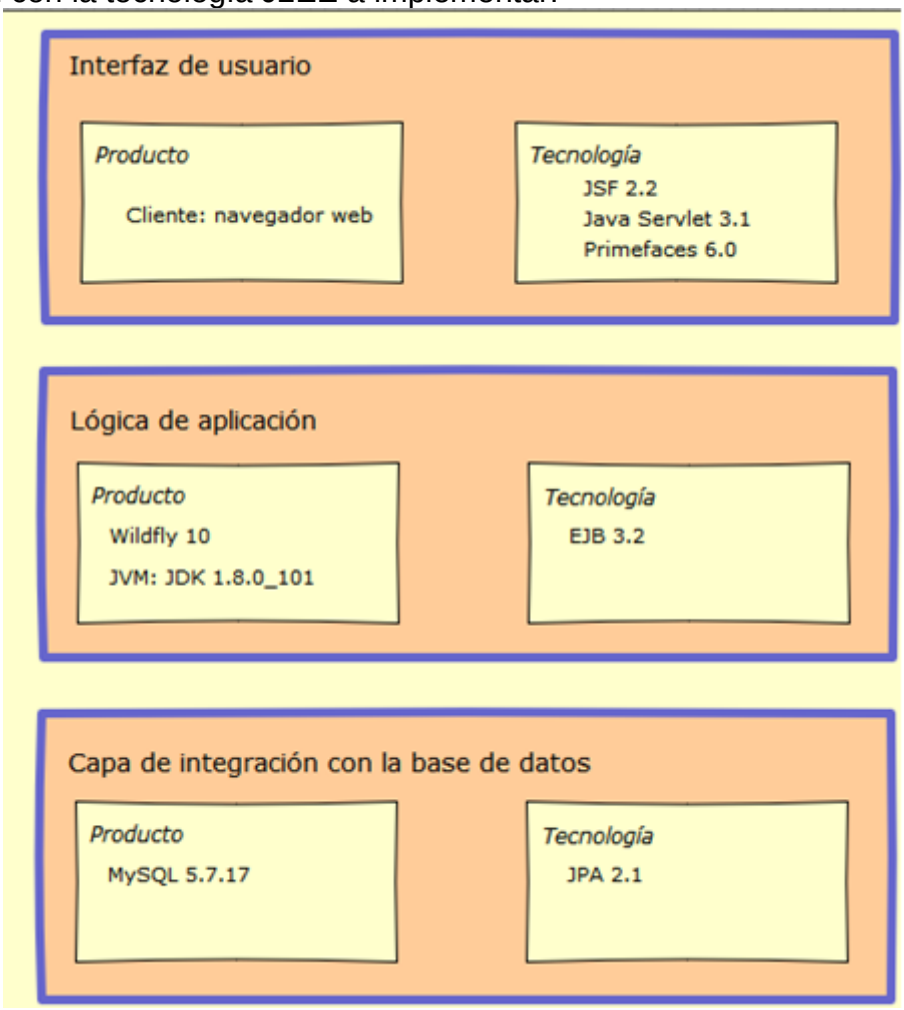
Detalle de interfaces en el componente TableManagement

3.3. Relación de la arquitectura de la aplicación con la tecnología J2EE

En cuanto a la tecnología J2EE se aplican los siguientes elementos en cada capa:

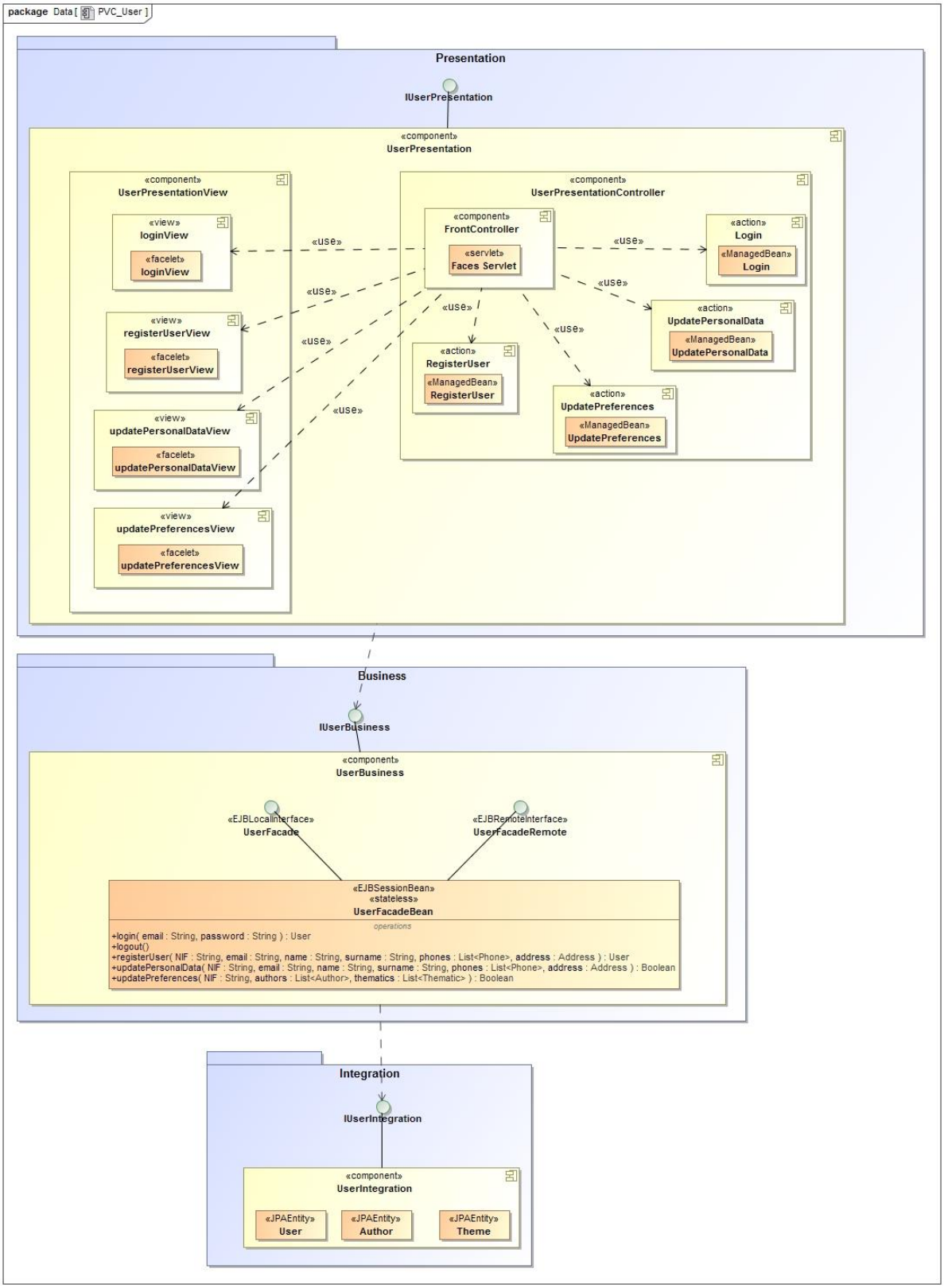
- Capa de presentación: en esta capa se aplica el patrón MVC (Modelo-Vista-Controlador). Las vistas se implementan mediante *facelets* (*JSF*), el controlador, que se encarga de gestionar el ciclo de vida de aplicaciones web que implementan la interfaz mediante *JSF*, se encuentra implícito en la tecnología J2EE (*Faces Servlet*) y el modelo se implementa mediante *Managed Bean*, cuya función es recoger la interacción del usuario con la interfaz (*facelets*). En esta capa se integra la tecnología J2EE de las vistas (JSF) con el framework Primefaces, con el objetivo de que la interfaz sea más atractiva y agradable para el usuario.
- Capa de negocio: en esta capa se implementa la lógica de negocio de la aplicación encapsulada en componentes EJB, que son componentes de servidor pertenecientes a la tecnología J2EE. En esta capa se integra también el framework Spring 4 (concretamente los módulos *EJB Integration* y *Transaction API*).
- Capa de integración: esta capa se encarga de hacer persistente la información en una base de datos. Se aplica la tecnología JPA, que permite representar un mapeo de objetos/relacional para administrar datos relacionales en aplicaciones Java (J2EE). Adicionalmente, proporciona un lenguaje de consultas propio (JPQL).

Se muestra a continuación la estructuración en capas de la aplicación, y su relación con la tecnología J2EE a implementar:

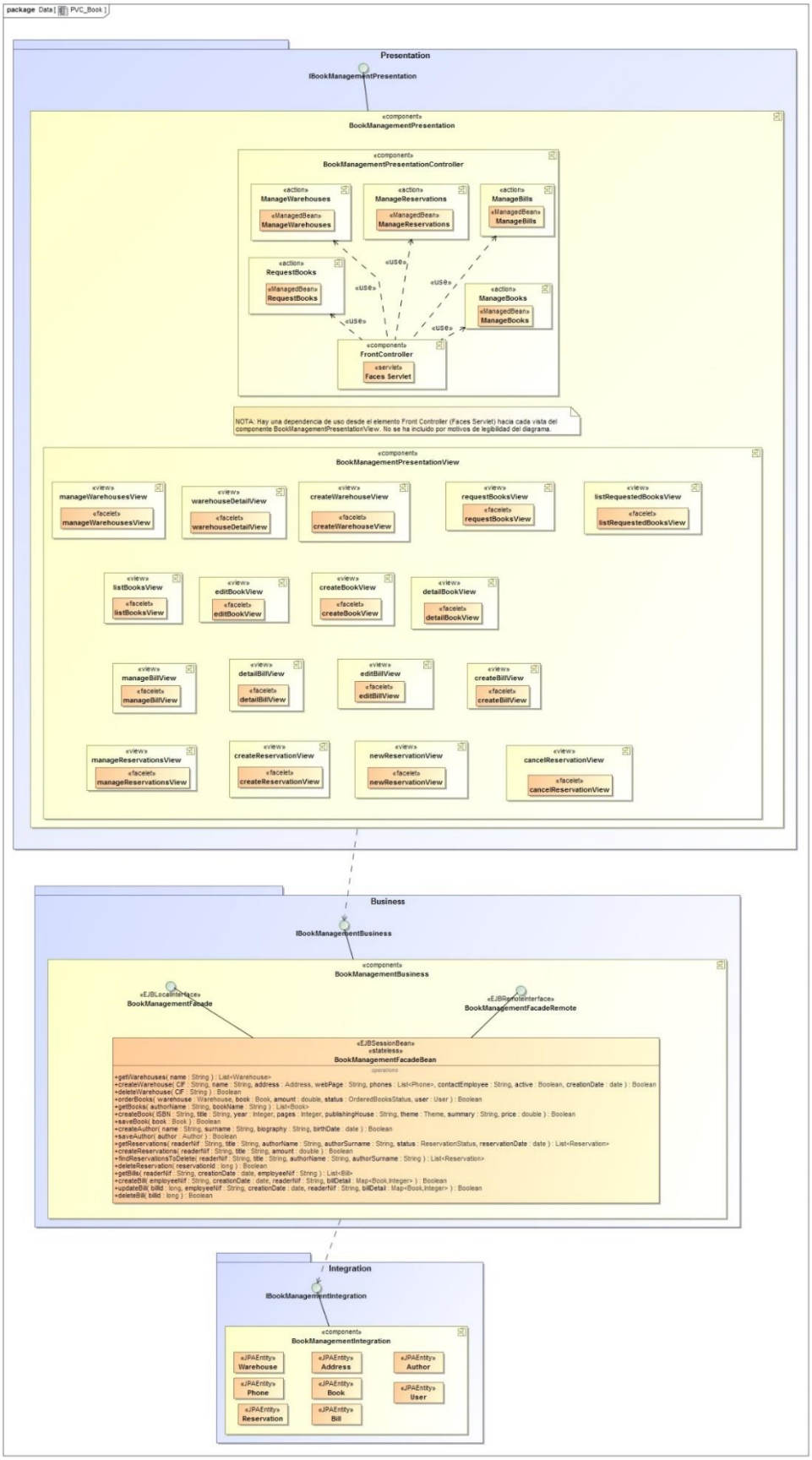


En este momento, es necesario comentar que aunque inicialmente en la capa de la lógica de la aplicación (capa de negocio) estaba previsto integrar J2EE con Spring, tras estudiar dicho Framework, se concluyó que en muchos aspectos EJB y Spring eran redundantes, además de que teniendo en cuenta el tiempo disponible para implementar el proyecto, el esfuerzo y dedicación necesarios para integrar Spring resultaba mayor que el aceptable, poniendo en riesgo la implementación de la aplicación.

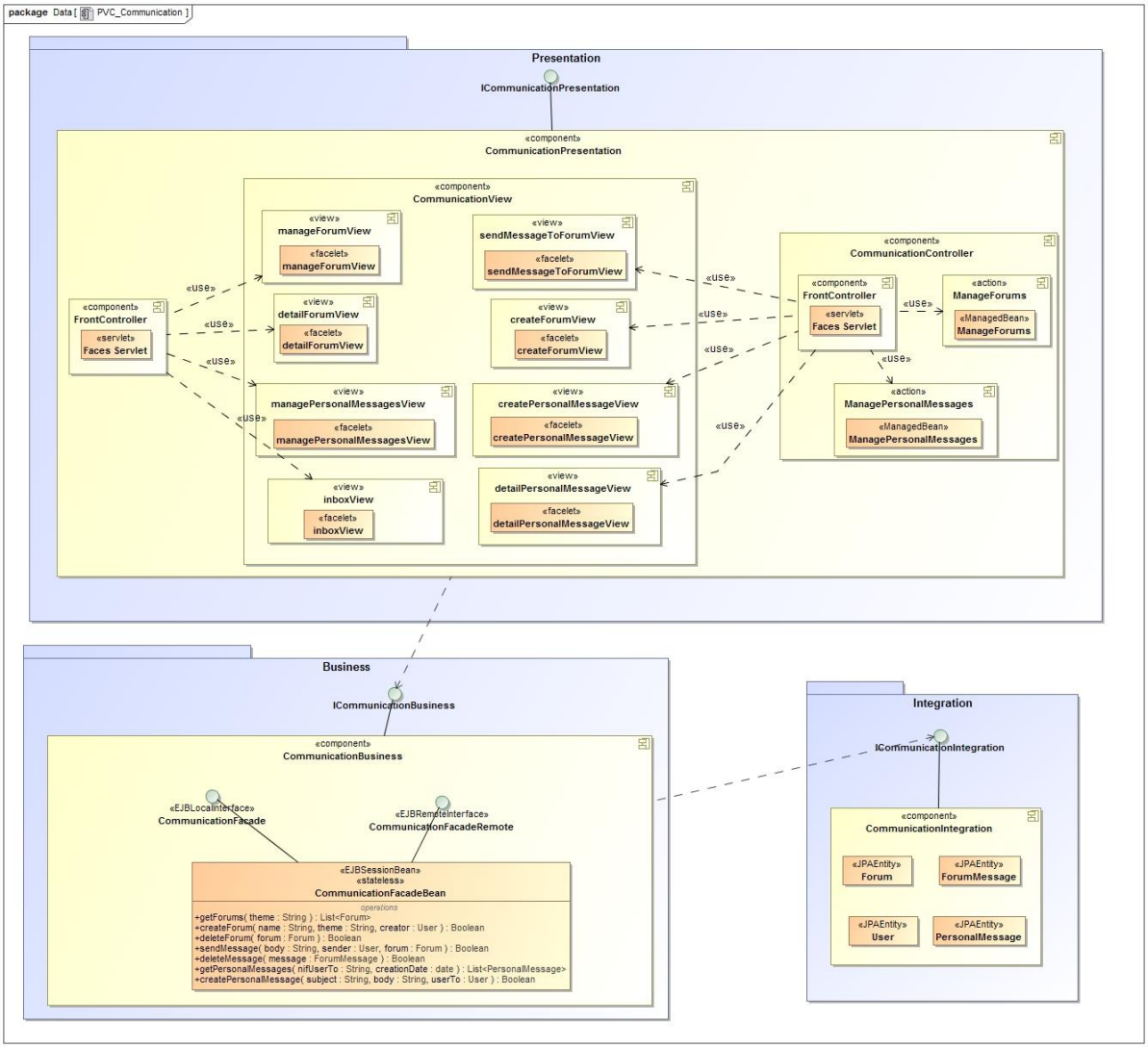
A continuación, se incluyen los diagramas de componentes, para cada componente en que se divide la aplicación, especificando las tecnologías integradas en cada capa:



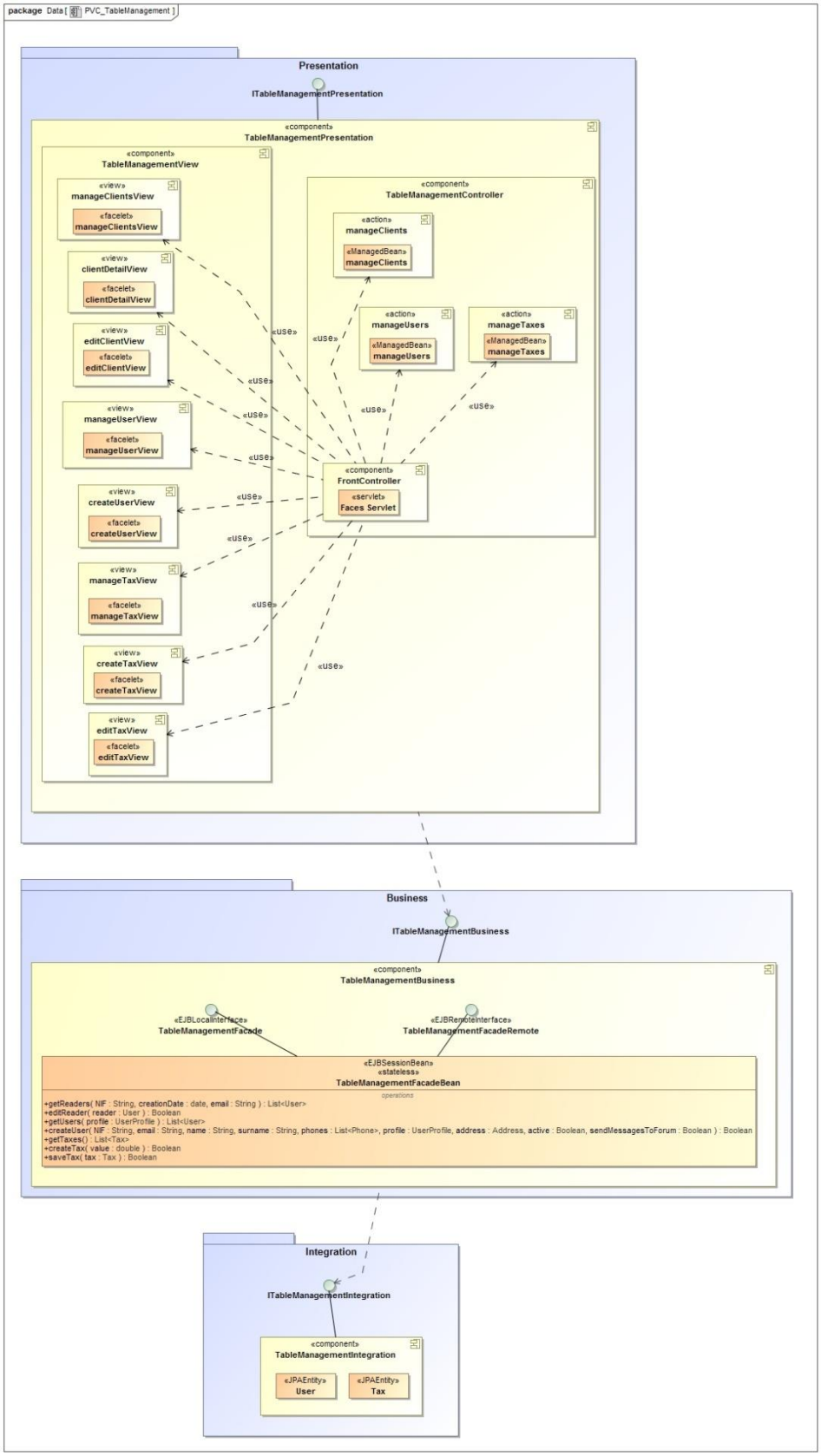
Componente Usuario



Componente Gestión de libros



Componente Comunicación



Componente Gestión de tablas

3.4. Diagramas de secuencia

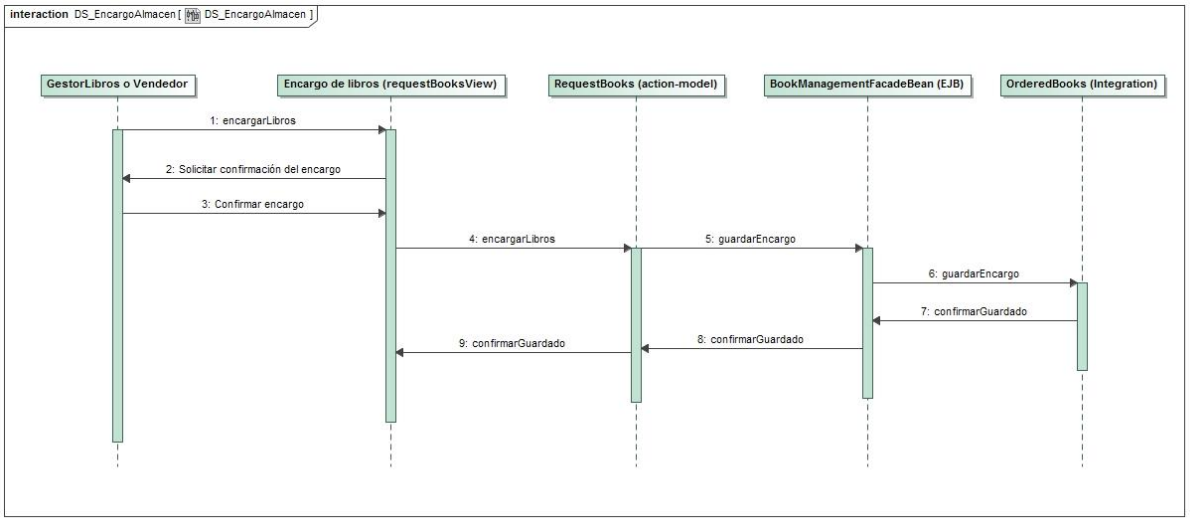
En este apartado se describen los diagramas de secuencia de aquellos casos de uso que se consideran más complejos.

De un modo complementario a la documentación textual y el diagrama de casos de uso, se decide realizar el diseño de los flujos principales de la aplicación, mediante diagramas de secuencia.

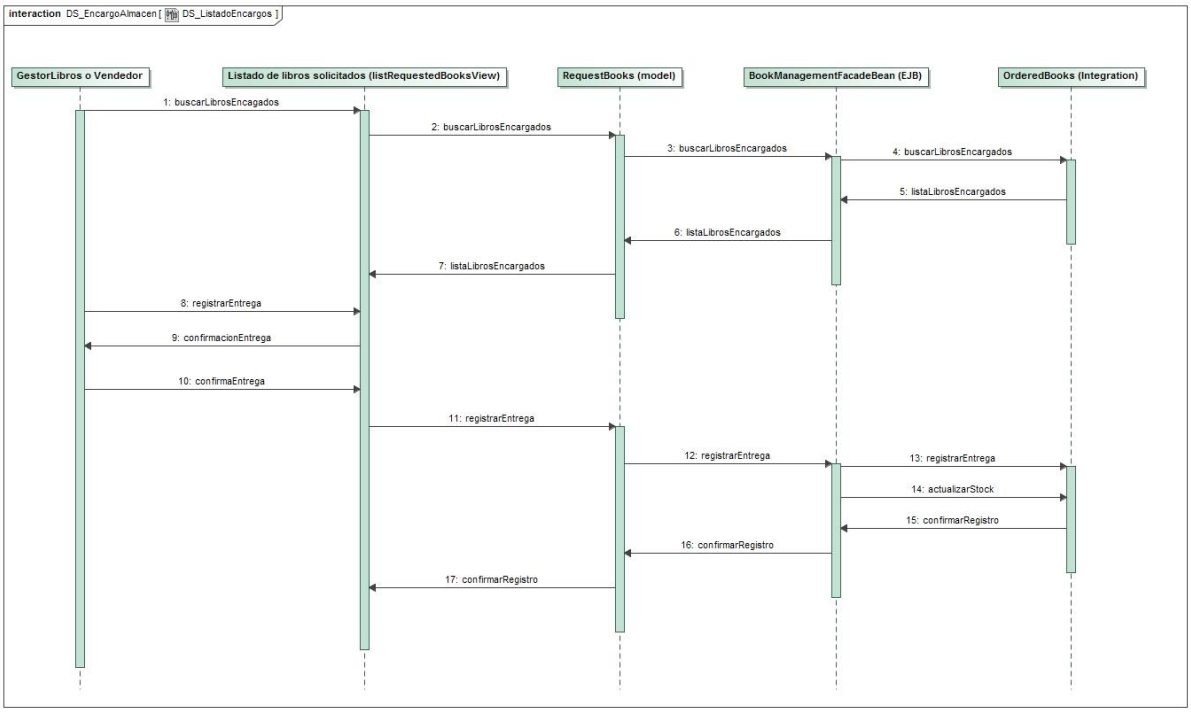
Con este objetivo, en una etapa previa al diseño de los diagramas de secuencia, se analiza, a partir de los casos de uso, cuáles de ellos se considera que componen el núcleo de la aplicación GLIB. Tras analizar los casos de uso se concluye que los principales son:

- Encargo de libros a almacén y listado de encargos
- Gestión de reservas: creación y anulación de reservas
- Gestión de facturas: creación, edición y eliminación de factura

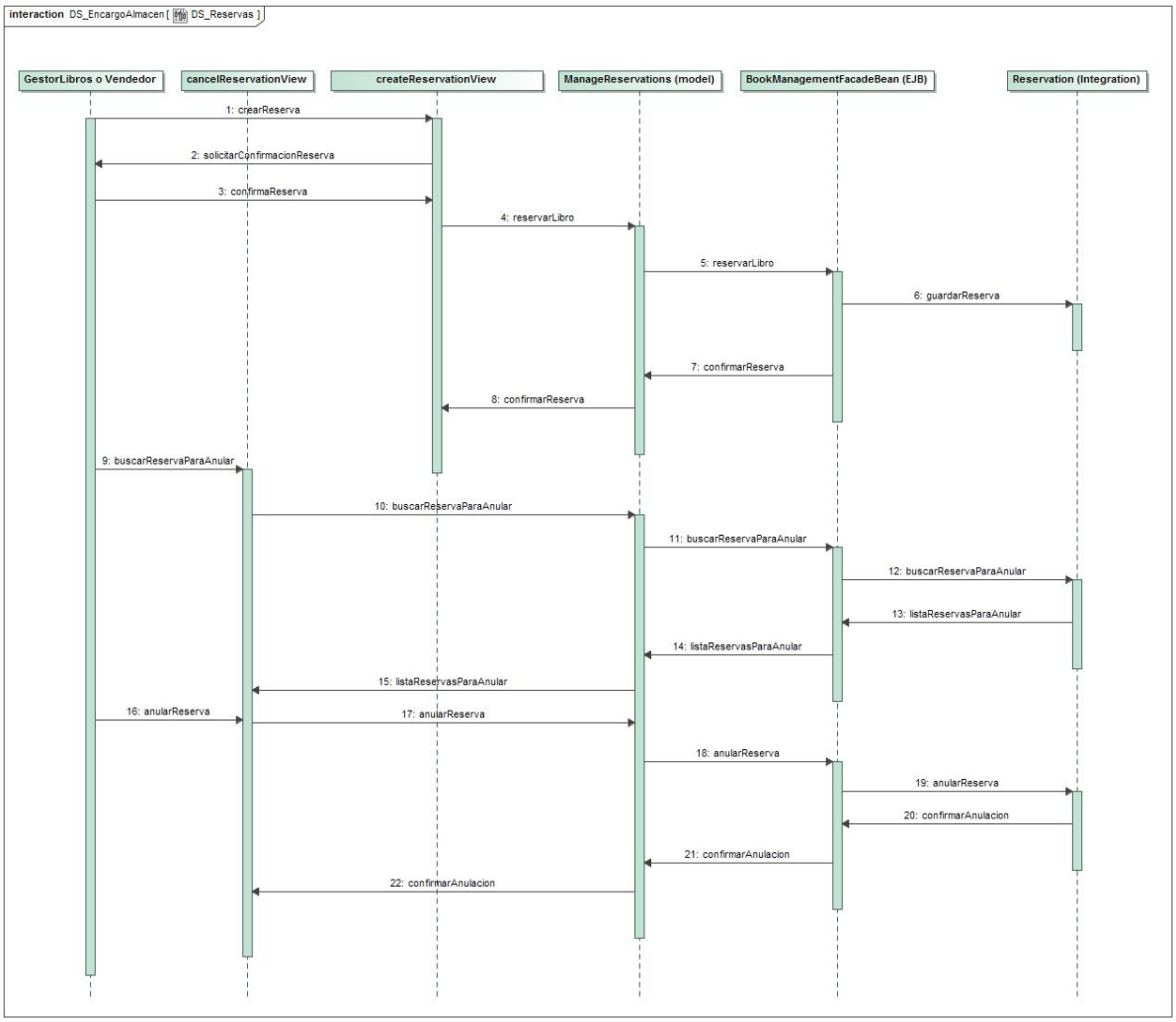
De esta forma, con estos tres casos de uso se cubren las funcionalidades del núcleo de la aplicación: solicitar libros a los almacenes para disponer de stock en la librería y poder responder a las peticiones de los clientes, permitir a los clientes la reserva de libros y, por último, la gestión de facturas, que permite a los empleados crear y gestionar las facturas para poder vender libros a los clientes.



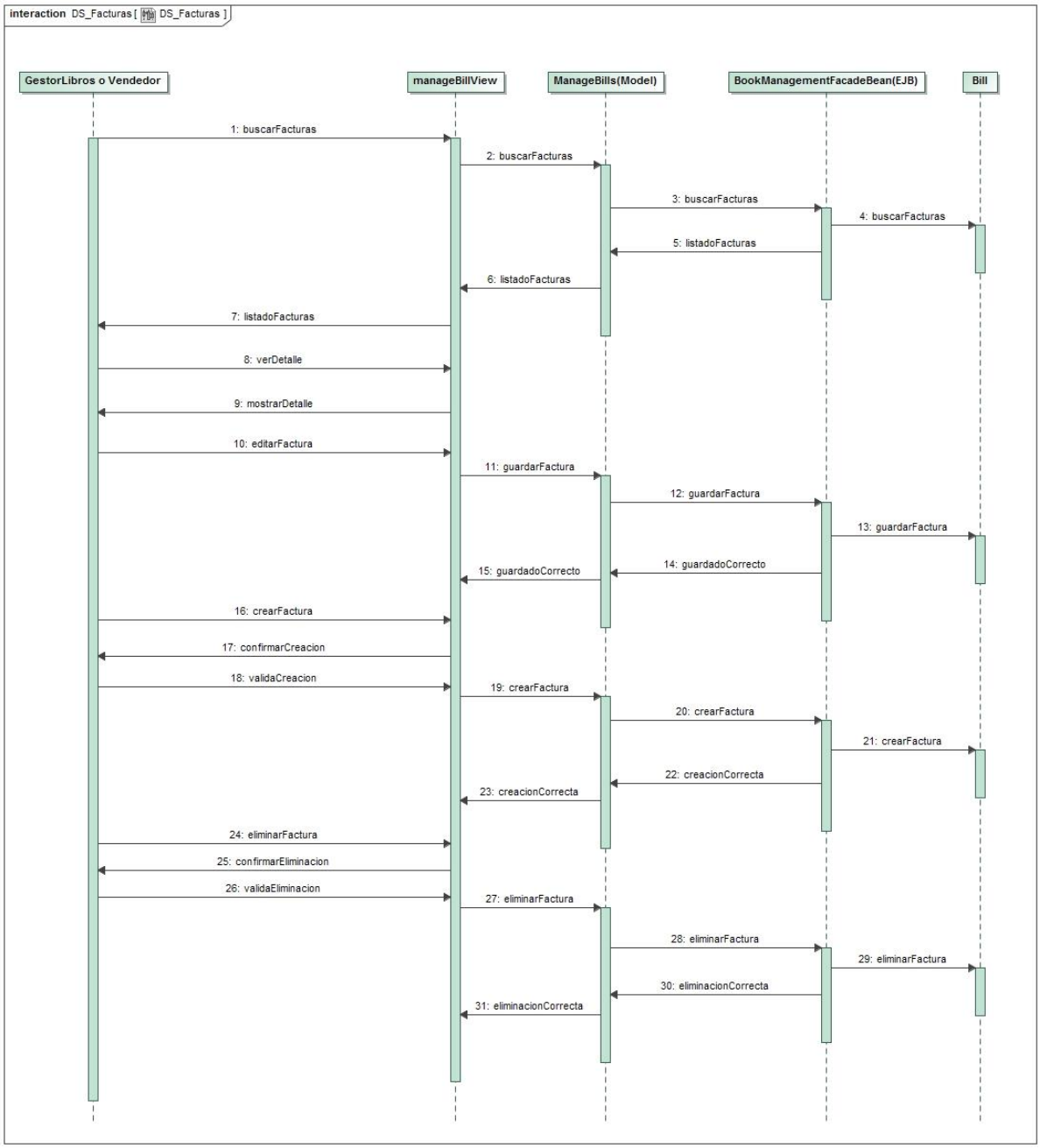
Encargo de libros a almacén



Listado de encargos y registro de entrega



Gestión de reservas



Gestión de facturas

3.5. Diseño relacional de la base de datos

A partir del diseño conceptual de la base de datos, como se comentó previamente en dicho apartado, se aplica la transformación del diseño conceptual al diseño relacional, verificando que se cumplen las formas normales.

Se incluye a continuación el diseño relacional:

User (nif, **name**, **surname**, **email**, **profile**, **active**, **sendMessagesToForum**, **forcePasswordChange**)

UserPhone (phone, user)

{user} es clave ajena de User
{phone} es clave ajena de Phone

UserAddress(user, address)

{user} es clave ajena de User
{address} es clave ajena de Address

Address (id, **address**, **number**, **zipCode**, **city**)

Phone (id, **number**)

PersonalMessage (id, **subject**, **body**, **creationDate**, **readed**, readedDate, **originUser**, **destinationUser**)

{originUser} es clave ajena de User
{destinationUser} es clave ajena de User

Author (id, **name**, **surname**, **biography**, **birthDate**)

Theme (id, **name**, **description**)

Book (id, **isbn**, **title**, **year**, **pages**, **publishingHouse**, **summary**, **price**, **discontinued**, **stock, theme**)

{theme} es clave ajena de Theme

UserBook (user, book)

{user} es clave ajena de User
{book} es clave ajena de Book

Warehouse (id, **name**, **cif**, web, **contactPerson**, **active**, **creationDate**, **deleted**)

WarehousePhone (phone, warehouse)

{warehouse} es clave ajena de Warehouse
{phone} es clave ajena de Phone

WarehouseAddress(address, warehouse)

{address} es clave ajena de Address
{warehouse} es clave ajena de Warehouse

Reservation (id, **amount**, **status**, **user**, **book**)

{user} es clave ajena de User
{book} es clave ajena de Book

OrderedBooks (id, **warehouse**, **book**, **amount**, **status**, **user**)

{warehouse} es clave ajena de Warehouse
{book} es clave ajena de Book
{user} es clave ajena de User

Bill (id, **creationDate**, **deleted**)

BillDetail (id, bill, **amount**, **book**)

{book} es clave ajena de Book
{bill} es clave ajena de Bill

BillInfo (**creationUser**, clientUser, bill)

{creationUser} es clave ajena de User
{clientUser} es clave ajena de User
{bill} es clave ajena de Bill

Forum (id, **name**, **theme**, **deleted**, **creator**)

{creator} es clave ajena de la relación User

ForumMessage (id, **body**, **deleted**, **sender**)

{sender} es clave ajena de la relación User

Posteriormente al diseño relacional, se analiza si el diseño conceptual cumple las tres primeras formas normales.

La primera forma normal (1FN), indica que cada relación (entidad en el diseño conceptual) si todos los atributos de la relación son atómicos, es decir, no son en sí mismos una relación, ni descomponible ni con multiplicidad de valores.

Se ha analizado cada relación del diseño relacional, y se concluye que todos los atributos de cada relación son atómicos.

La segunda forma normal (2FN) establece que una relación la cumple si, y solo si, está en primera forma normal y todo atributo que no forma parte de una clave candidata depende completamente de todas las claves candidatas de la relación.

Se ha analizado cada relación del diseño relacional, y se concluye que cada atributo (en cada relación) depende completamente de todas las claves candidatas de la relación.

Una relación está en tercera forma normal (3FN) si, y solo si, está en segunda forma normal y ningún atributo que no forma parte de una clave candidata depende de un conjunto de atributos que contiene alguno que no forma parte de una clave candidata.

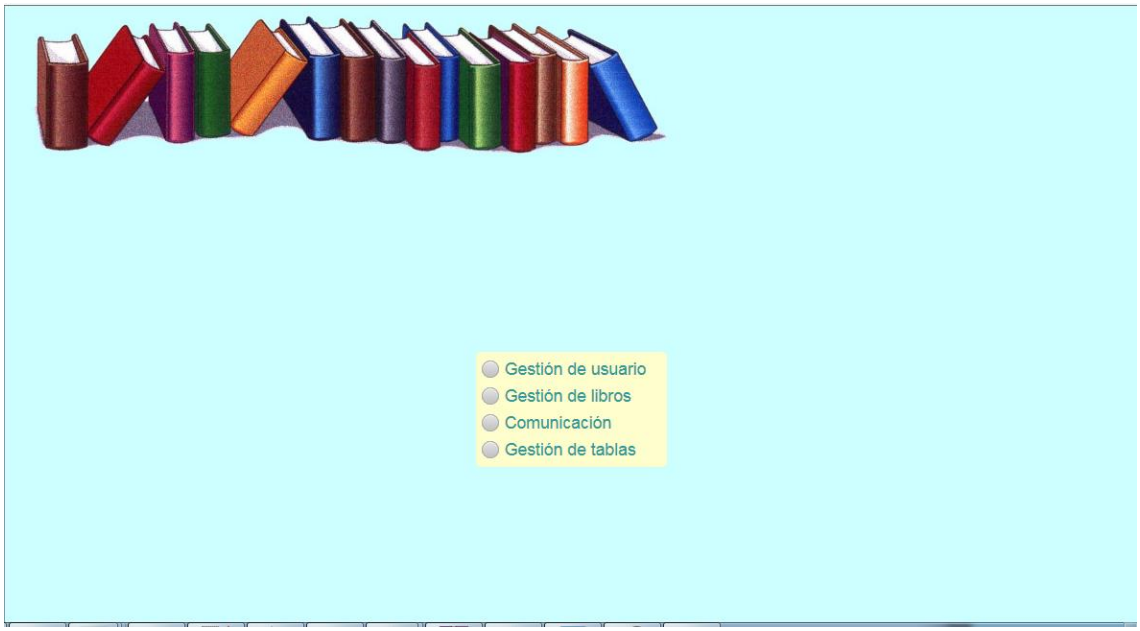
Se han analizado todas las relaciones del diseño relacional y se concluye que en ninguna de ellas hay algún atributo que dependa de un atributo o de un conjunto de atributos, por lo que cada relación cumple la 3FN.

Tras analizar las tres primeras formas normales, se puede concluir que el diseño relacional cumple la tercera forma normal.

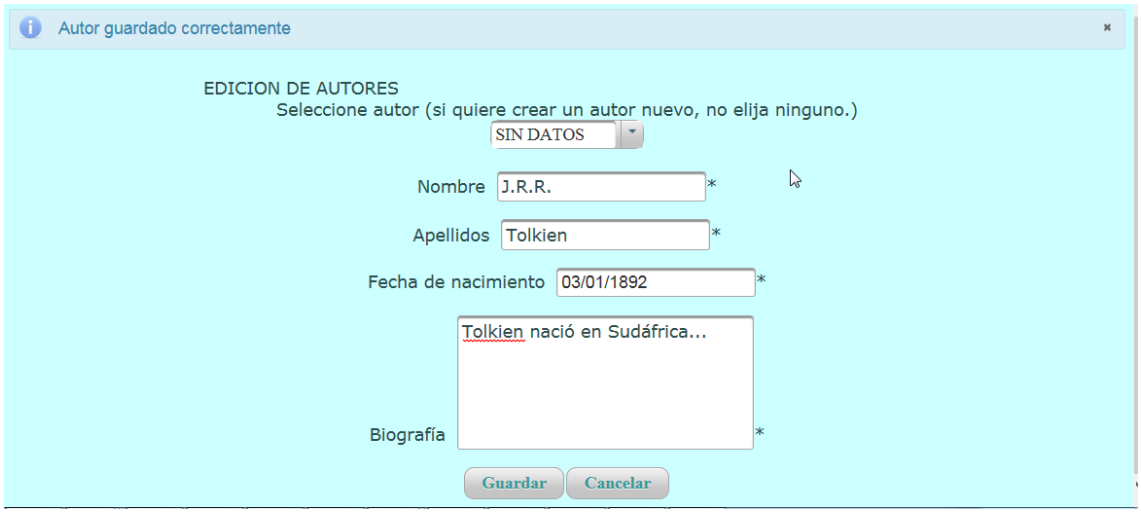
3.6. Interfaz gráfica de usuario

Se incluyen imágenes de las principales funcionalidades de la aplicación.

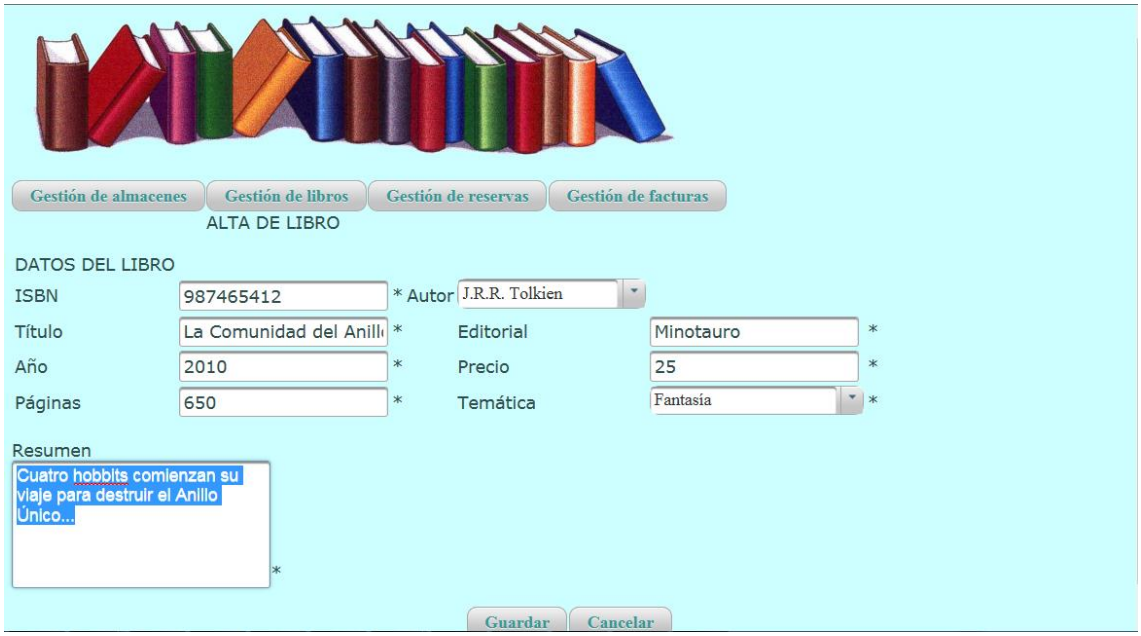
Inicio de sesión



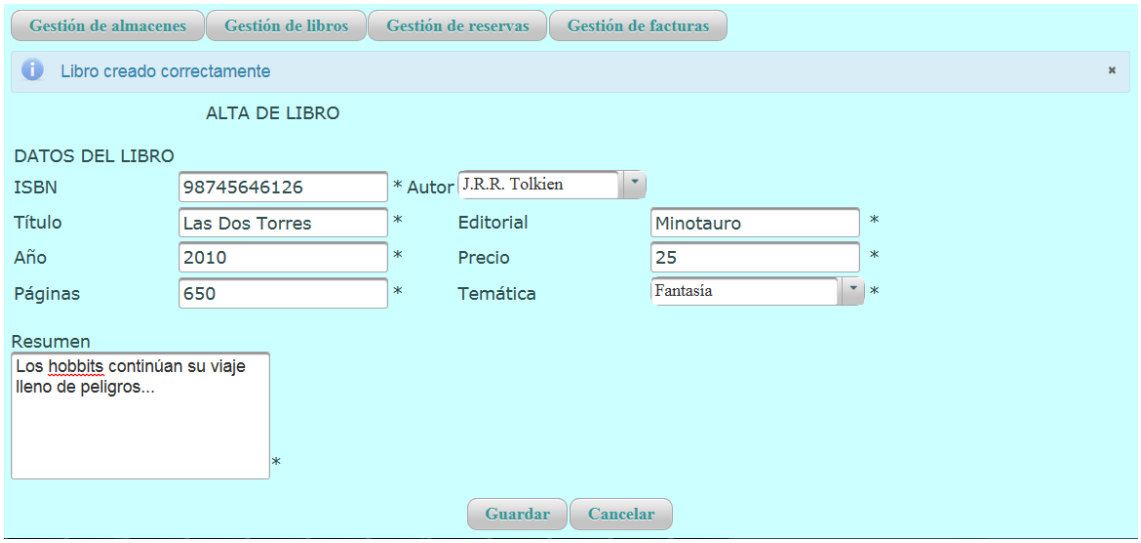
Pantalla principal del perfil Gestor Libros



Edición de autores: perfil Gestor Libros



Alta de libros: perfil Gestor Libros



Alta de libros: perfil Gestor Libros

Listado de libros: perfiles Gestor Libros, Vendedor y Lector

Edición de libro

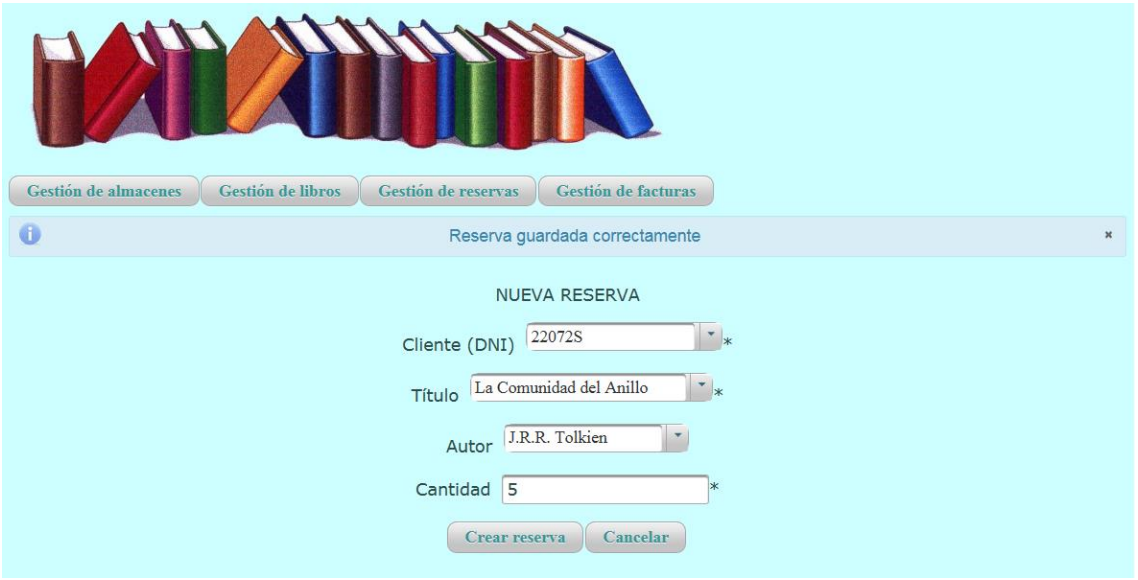
Alta de almacén: perfil Gestor Libros

Listado de almacenes: perfiles Gestor Libros y Vendedor

Encargo de libros: perfiles Gestor Libros y Vendedor

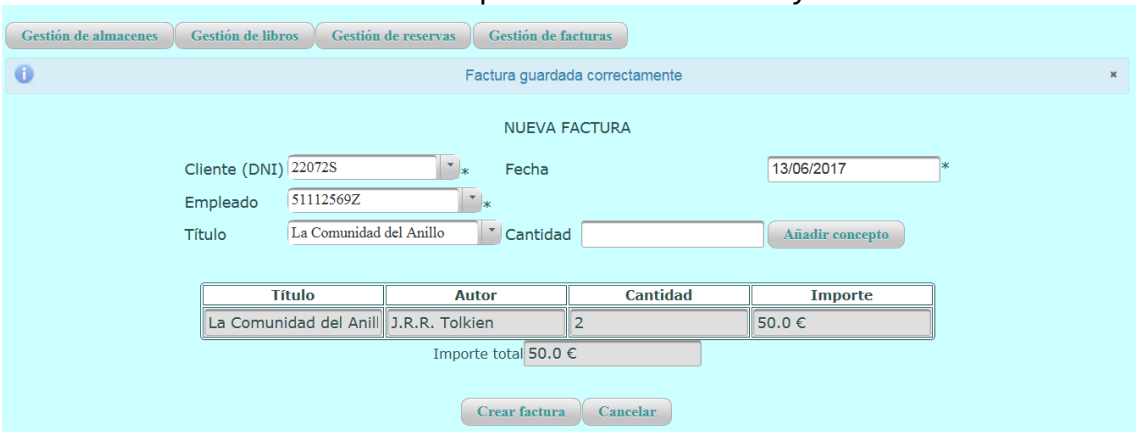
Listado de encargos: perfiles Gestor Libros, Vendedor

Listado de encargos: perfiles Gestor Libros, Vendedor (registro de entrega)

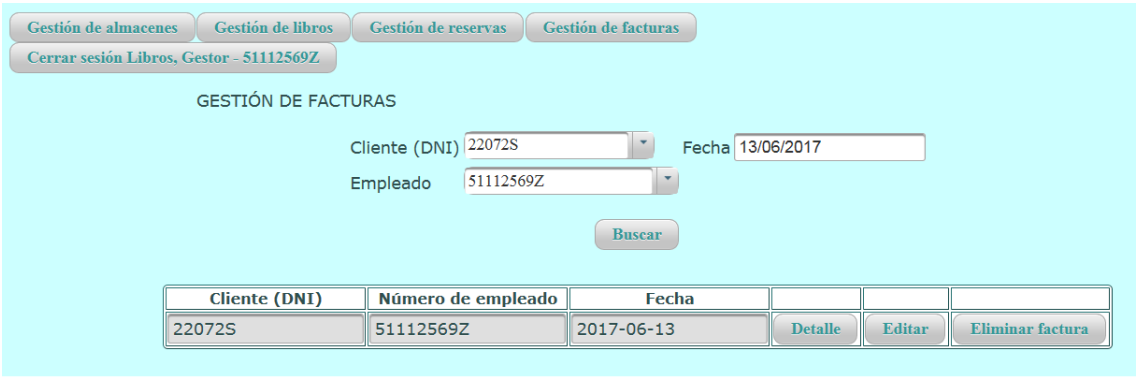


Reserva de libros: perfil Gestor Libros y Vendedor

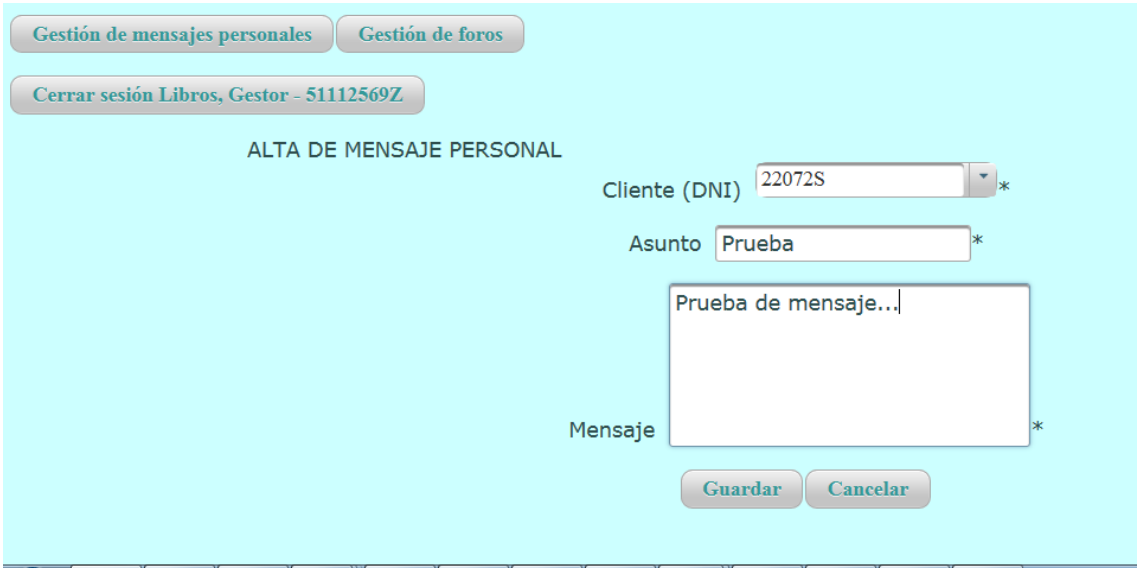
Listado de reservas: perfiles Gestor Libros y Vendedor



Alta de factura: perfiles Gestor Libros y Vendedor



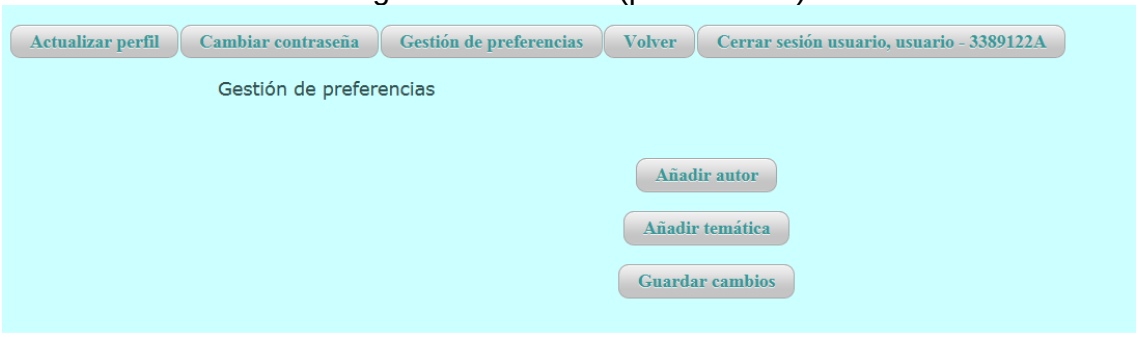
Listado de facturas: perfiles Gestor Libros y Vendedor. En el caso del perfil Lector únicamente permitiría elegir fecha.



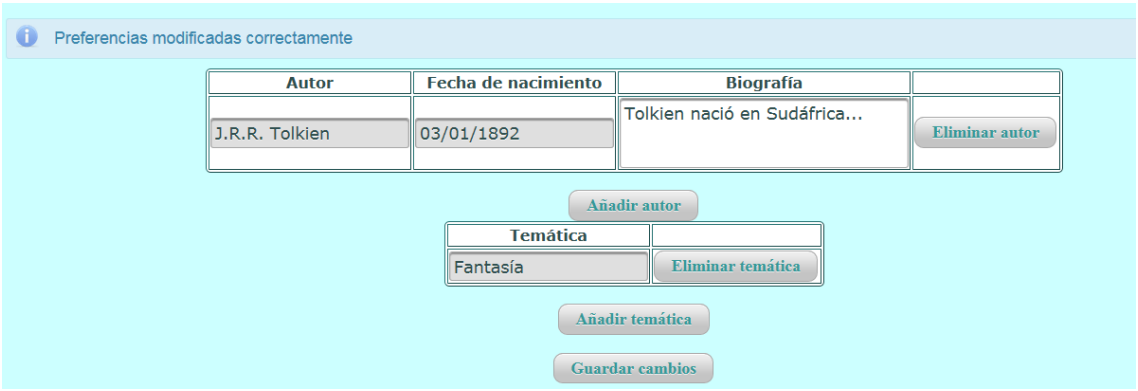
Alta de mensaje personal: perfil Gestor Libros

Listado de mensajes personales: perfiles Gestor Libros y Vendedor. En el caso del perfil Lector, no permitiría elegir DNI, sólo fecha.

Registro de usuario (perfil Lector)



Gestión de preferencias (perfil Lector)



Gestión de preferencias (perfil Lector)

4. Implementación

El proyecto Java se estructura en los siguientes componentes:

- src: carpeta que contiene el código fuente de las clases Java, interfaces Java y enumerados que requiere la aplicación. Las subcarpetas o paquetes que contiene son:
  - ejb: este paquete contiene las clases EJB, que forman la capa de negocio de la aplicación.
  - enums: este paquete contiene los enumerados utilizados en la aplicación.
  - jpa: este paquete contiene las clases Java que representan las entidades JPA, constituyendo la capa de integración.
  - managedbean: este paquete contiene las clases java que forman parte de la capa de presentación, recogiendo la interacción del usuario con la interfaz de la aplicación.
  - utils: este paquete contiene diversas clases Java de utilidad general en la aplicación.
  - META-INF: esta carpeta contiene los ficheros application.xml y persistence.xml, donde el primero configura cómo se estructura la aplicación al compilarse para el servidor de aplicaciones, y el segundo configura las opciones de persistencia de la base de datos, logrando exportar el esquema en la base de datos creando las tablas o validando el esquema de la base de datos.
- docroot: esta carpeta contiene los recursos necesarios para la interfaz de usuario (imágenes y hojas de estilo), así como los ficheros xhtml que definen las distintas páginas con las que interactúa el usuario. Las subcarpetas son:
  - resources: contiene los recursos necesarios, en las siguientes carpetas
    - css: hojas de estilo.
    - Img: imágenes de la aplicación.
  - WEB-INF
    - Lib: contiene la librería del framework PrimeFaces, versión 6.1, requerida en la interfaz de la aplicación.

        - faces-config.xml: fichero que configura las opciones necesarias para JSF.
        - web.xml: fichero con las opciones de configuración para el despliegue de la capa de presentación de la aplicación, donde por ejemplo se indica el controlador Faces Servlet, o cuál es la página de inicio de la aplicación.
- Librerías: se requieren la librería jdk 1.8 y el entorno de ejecución del servidor Wildfly versión 10, que se integra en Eclipse.

4.1. Vistas

Las vistas se han definido de tal forma que se dispone de la vista principal de la aplicación (mainView.xhtml), que constituye el punto de entrada a la misma.

El resto de vistas se han definido para implementar las funcionalidades de la aplicación. Se ha procurado seguir el diseño original de la aplicación al definir las vistas, aunque en algunos casos se ha requerido alguna vista adicional o se ha considerado que resultaba más adecuado otro nombre para la vista.

En este apartado cabe destacar en primer lugar el hecho de haber definido una plantilla (headerView.xhtml), que contenía elementos comunes a todas las vistas (el color de fondo, la imagen de la parte superior), de tal forma que se puede considerar como “reutilización de código”, salvando las distancias con la herencia de clases. Esta aproximación ha resultado muy útil ya que al cambiar la imagen en la plantilla se cambió en todas las páginas de la aplicación.

Cabe mencionar también que el framework PrimeFaces, aunque inicialmente resultó costoso que se integrase con JSF, sin duda merece la pena porque proporciona una interfaz mucho más amigable para el usuario y dispone de elementos de interfaz web con funcionalidades muy atractivas y que resuelven situaciones que con JSF resultan laboriosas. Ha resultado una experiencia enriquecedora, especialmente al permitir definir menús en la aplicación o mostrar al usuario mensajes (cuadros de diálogo).

4.2. Modelo (Managed Bean)

El modelo, perteneciente a la capa de presentación, se implementa mediante clases Java (managed bean), que se comunican, a través del controlador implícito en la tecnología JSF (Faces Servlet), recogiendo la interacción del usuario con la interfaz de la aplicación, y respondiendo al mismo con las acciones realizadas.

La implementación de estas clases ha resultado muy útil también, ya que se ha logrado profundizar en una mejor interacción entre las vistas y el modelo, haciendo más eficiente, y permitiendo, la reutilización de vistas entre los diferentes perfiles que componen la aplicación. Por ejemplo, en función del perfil del usuario que ha iniciado sesión, en una vista se mostraban unos elementos o no, es decir, al perfil Gestor Libros se le permitía la edición mientras que al perfil Vendedor no, siendo la vista idéntica excepto en que se permita editar o no.

En esta capa se han definido las clases Java previstas, que se muestran a continuación:

- Login: clase Java que recoge la interacción del usuario cuando inicia sesión en la aplicación.
- ManageBills: clase Java que recoge la interacción del usuario al crear facturas, modificar facturas, eliminarlas o listar las facturas.
- ManageBooks: clase Java que recoge la interacción del usuario al crear o editar libros, crear o editar autores, listar libros, etc.
- ManageForums: clase Java que recoge la interacción del usuario al crear o eliminar foros, al enviar mensajes a foros y al eliminar mensajes.
- ManagePersonalMessages: clase Java que recoge la interacción del usuario al crear mensajes personales, al listar mensajes personales enviados a usuarios o al consultar la bandeja de entrada.
- ManageReservations: clase Java que recoge la interacción del usuario al listar reservas de libros, o al crear, anular o editar reservas.
- ManageWarehouses: clase Java que recoge la interacción del usuario al crear o editar almacenes, al eliminar almacenes y al listar almacenes.
- RegisterUser: clase Java que recoge la interacción del usuario al registrarse en la aplicación.
- RequestBooks: clase java que recoge la interacción del usuario al encargar libros a almacenes, o consultar encargos de libros ya realizados.
- TableManagement: clase Java que recoge la interacción del usuario en aquellos casos de uso relacionados con la gestión de tablas: listado de clientes, edición o creación de clientes, creación de usuarios de diferentes perfiles, gestión de impuestos.
- UpdatePersonalData: clase java que recoge la interacción del usuario en la actualización de datos personales.
- UpdatePreferences: clase Java que recoge la interacción del usuario, de perfil Lector, al actualizar sus preferencias (autores y temáticas).

Adicionalmente a los managed bean previstos, se han definido los siguientes:

- ApplicationMenu: clase Java que define los títulos a mostrar en las vistas.
- BookManagerMain: clase Java que recoge la interacción de usuarios de perfil GestorLibros, tras acceder a la aplicación, eligiendo a qué módulo quiere acceder.
- ReaderMain: clase Java que recoge la interacción de usuarios de perfil Lector, tras acceder a la aplicación, eligiendo a qué módulo quiere acceder.
- SellerMain: clase Java que recoge la interacción de usuarios de perfil Vendedor, tras acceder a la aplicación, eligiendo a qué módulo quiere acceder.

4.3. Capa de negocio (EJB)

La implementación de esta capa ha permitido profundizar en el desarrollo de EJB y su interacción con la capa de integración, logrando que la información persistida en base de datos sea íntegra y consistente.

En esta capa, se han definido los EJB previstos en el diseño, aunque en algunos de ellos ha sido necesario definir métodos no previstos, que se mencionan a continuación, organizados por EJB:

- BookManagementFacadeBean / BookManagementFacadeRemote / BookManagementFacadeLocal
  - findWarehouseById: fue necesario añadir este método para uso interno en esta clase.
  - saveWarehouse: fue necesario definir este método ya que la idea inicial era utilizar el método createWarehouse tanto para crear como para guardar almacenes, pero la lógica no es idéntica en los dos casos, por lo que se creó este método para guardar almacenes.
  - getAuthors: este método no estaba previsto en el diseño incial, siendo necesario para obtener los autores, utilizado en varias clases del modelo de la capa de presentación (managed bean).
  - getAuthor: este método fue necesario crearlo para dar soporte a algunas funcionalidades.
  - createAuthor: este método se ha llamado saveAuthor.
  - getThemes, getTheme: estos métodos se han creado para obtener temáticas, que no estaba previsto en el diseño inicial, pero es necesario en varias funcionalidades de la aplicación, como edición o creación de libros, gestión de preferencias del perfil Lector,…
  - getAuthorBooks: este método obtiene los libros asociados a un autor, sin estar previsto en el diseño inicial pero siendo necesario en la aplicación, para mostrar al usuario los libros escritos por un autor.
  - getOrderedBooks: este método no estaba previsto en el diseño inicial, siendo necesario para mostrar el listado de libros encargados a almacenes.
  - registerOrderedBook: este método equivaldría a orderBooks del diseño inicial, ya que en la aplicación se ha implementado para permitir el encargo de un libro cada vez.
  - deliverReservation
  - findReservationsToDelete: este método, aunque estaba previsto en el diseño inicial, se considera que no es necesario ya que el método *getReservations*, permite obtener aquellas reservas de libros con un estado concreto, que sería la lógica del método *findReservationsToDelete* puesto que las reservas de un estado concreto son aquellas que se pueden eliminar.

    - deleteBill: inicialmente se diseñó este nombre, pero en la implementación se decidió que el nombre *cancelBill* representaba mejor el objetivo del método.
    - getBill: este método fue necesario definirlo para uso interno en esta clase y para la edición de facturas.
- UserFacadeBean / UserFacadeRemote / UserFacadeLocal
    - updatePassword: fue necesario definir este método para permitir únicamente el cambio de contraseña.
    - updateFirstAccessPassword: este método surgió a partir de la idea obtenida durante la implementación, de obligar a cambiar la contraseña a aquellos usuarios creados desde la aplicación, puesto que al crear un usuario por defecto se le asigna la contraseña igual al usuario (email), por lo que es recomendable cambiarla en el primer acceso.
    - getReaderUsers: este método fue necesario para obtener los usuarios de perfil Lector.
    - getEmployees: este método fue necesario para obtener los usuarios de perfil Gestor Libros y Vendedor.
    - getUser(Email): este método obtiene un usuario a partir del email, necesario para la actualización de datos personales.
    - getUserByNif: este método fue necesario para varias funcionalidades como la gestión de mensajes personales y la creación de reservas y de facturas.
- CommunicationFacadeBean / CommunicationFacadeRemote / CommunicationFacadeLocal
    - getForumMessages: este método se definió para obtener directamente los mensajes asociados a un foro, mostrándolos en el detalle de un foro.
    - sendMessageToForum: este método corresponde al método sendMessage del diseño, se decidió modicar el nombre por ser demasiado parecido al método createMessage, de esta forma se distingue perfectamente el objetivo de cada uno.
    - deletForumMessage: en este caso, al igual que el anterior el nombre podría confundirse con mensajes de tipo personal, por lo que se decidió renombrarlo. En el diseño original era deleteMessage.
    - getMessages: este método se decidió renombrar del nombre en el diseño original: getPersonalMessages, ya que se tiene claro que los mensajes sin “apellido” son de tipo personal.
    - saveMessage: este método corresponde al método createPersonalMessage del diseño original.
- TableManagementFacadeBean / TableManagementFacadeRemote / TableManagementFacadeLocal
    - createTax, saveTax: estos dos métodos del diseño original se han definido en la implementación como saveTax.
    - getTaxById: este método, no previsto en el diseño original, fue necesario para la edición de impuestos.
    - saveUser: este método, no previsto en el diseño original, fue necesario para la edición de usuarios en el módulo Gestión de Tablas.

- saveUserToActivateOrDelete: este método, no previsto en el diseño original, permite activar o dar de baja usuarios, necesario para la funcionalidad activación y baja de usuarios.
- getUsersByProfileAndOrNif: este método, no previsto en el diseño original, fue necesario para mostrar usuarios en las funcionalidades de listado de usuarios y de gestión de clientes, filtrando por perfil y/o NIF.
- getUsers: este método del diseño original se ha renombrado a getUsersByProfile, que se considera más autodescriptivo.
- getUser (nif, email): este método, no previsto en el diseño original, fue necesario crearlo para la gestión de clientes.
- getUserByNif: este método, no previsto en el diseño original, fue necesario crearlo para funcionalidades como la edición o el detalle de clientes, la carga de usuarios a activar o dar de baja, y el guardado de usuarios.

4.4. Capa de integración (JPA)

La capa de integración se ha implementado según lo previsto en el diseño.

Se mencionan dos diferencias importantes, que se han estimado necesarias ya que para usuarios y almacenes bastaría con un teléfono de contacto, al disponer de un correo electrónico a modo de comunicación.

En consecuencia, en las entidades JPA UserJPA y WarehouseJPA la colección *phones* no es necesaria, siendo sustituida por un atributo de tipo cadena, *phone*, que contiene el teléfono de contacto.

En la entidad AddressJPA, que representa las direcciones, en el diseño se implementó el atributo *number*, que representaba el número de la calle, sin embargo se ha considerado mejor incluirlo dentro del atributo address (que en el caso de la clase AddressJPA se denomina *name*), por facilidad del usuario al introducir únicamente la dirección, la ciudad y el código postal.

4.5. Elementos adicionales

En el proyecto, se han definido las siguientes clases Java, como apoyo a otros elementos del proyecto (paquete utils):

- Helper.java: esta clase contiene utilidades de uso general, como codificación y decodificación de contraseña de usuario (necesario en el registro de usuarios, en el login, al guardar usuarios, …), validación de email (al crear usuarios, al crear o actualizar almacenes, …), validación de que las contraseñas cumple que sean iguales, y que cumplen los requisitos de complejidad necesarios), validación de que el NIF de usuarios tiene el formato correcto, así como métodos de tratamiento de fechas.
- RequestBooksTableRow.java: esta clase encapsula la funcionalidad necesaria para encargar libros a almacenes, siendo el objetivo permitir

encargar cantidades de libros diferentes en una sola petición a la capa de negocio.

5. Objetivos conseguidos

En primer lugar, se abordan los objetivos del desarrollo de la aplicación web para gestionar una librería. En este ámbito cabe destacar que se ha logrado, con el tiempo disponible, desarrollar una aplicación web que representa el núcleo de la aplicación web, siendo posible mejorarla añadiendo funcionalidades adicionales, que mejoran la experiencia del usuario, sobretodo del cliente (el lector). Respecto a los requisitos funcionales, la meta en la etapa de desarrollo, ha sido implementar aquellas funcionalidades que se consideran el núcleo de la aplicación: el perfil Gestor de Libros (gestión de almacenes, gestión de libros y autores, gestión de reservas, gestión de facturas, gestión de clientes, creación y edición de usuarios, y comunicación) y el perfil Lector (gestión del perfil de usuario: datos personales, gestión de preferencias y cambio de contraseña; gestión de reservas; consulta de libros; comunicación).

En segundo lugar, se evalúan los objetivos en cuanto al aprendizaje tecnológico, que es el verdadero objetivo del TFG, la aplicación web constituye una idea con la que profundizar en la tecnología J2EE. En este aspecto, se considera que el TFG ha permitido ahondar en la tecnología J2EE, logrando una aplicación en la que se procura aplicar la reutilización de código lo máximo posible, aplicando marcos de trabajo orientados a aplicaciones de entorno empresarial, con la implementación de patrones de diseño (como MVC), y enfocadas a entornos distribuidos.

Se ha podido profundizar en la interacción entre las vistas y el modelo, y de éste último con la capa de negocio (EJB), así como en las entidades JPA y el lenguaje JPQL.

Por otro lado, se ha aprendido desde cero el marco de trabajo o *framework PrimeFaces*, que proporciona características que JSF por sí solo no dispone, haciendo la interfaz mucho más atractiva y amigable para el usuario, y permitiendo aplicar *widgets* que en JSF son complicados de conseguir, por no decir imposible. Especialmente en PrimeFaces se ha valorado mucho la propiedad *rendered* de sus elementos de interfaz, ya que permite reutilizar vistas entre diferentes perfiles, mostrando o habilitando ciertos elementos de la vista en función del perfil del usuario que accede a la vista. También se ha valorado en PrimeFaces los elementos “<p:messages/>” y “<p:growl”, que permiten mostrar mensajes al usuario, proporcionando feedback de una manera intuitiva y agradable a la vista. Se ha aplicado el estilo css3, que también ayuda a que la interfaz sea más atractiva.

A modo de reflexión considero que la tecnología J2EE permite desarrollar aplicaciones que proporcionan un marco de trabajo fácil de comprender y usar para desarrollar aplicaciones, a la par que implementa patrones de diseño y se ajusta a diversos entornos con requisitos funcionales y no funcionales diversos, como puede ser sistemas distribuidos, por lo que continuaré avanzando en el conocimiento de esta tecnología.

6. Trabajo futuro y posibles mejoras

6.1. Mejoras en la aplicación

6.1.1 Perfil Lector

En este perfil se considera que podrían añadir las siguientes funcionalidades:

- Una funcionalidad que permitiese a un usuario en su perfil subir imágenes, que pudiese compartir con el resto de usuarios.
- Una funcionalidad que permitiese a dos usuarios intercambiar mensajes a través de la aplicación, aprovechando la funcionalidad desarrollada para el perfil Gestor Libros, que envía mensajes usuarios de perfil Lector.
- Una funcionalidad para permitir hacer compras online, incluyendo el seguimiento del estado del pedido, pasarelas de pago, etc. Esta funcionalidad también afectaría a los perfiles Gestor Libros y Vendedor, que actualizarían el estado de las compras online (Recibido, Preparado, Enviado, etc.).

6.1.2 Perfil Almacén

Se crearía este perfil nuevo para los almacenes a los que se encargasen libros. El objetivo es que los almacenes pudiesen comunicar el estado de los pedidos que realizasen las librerías. Las funcionalidades serían las siguientes:

- Gestión de usuario: cambio de contraseña y actualización de datos del almacén.
- Gestión de solicitudes: módulo para gestionar los pedidos de las librerías, permitiendo actualizar el estado de las solicitudes.

6.1.3 Perfil lector

Al registrarse un usuario (de perfil Lector) desde la aplicación web, se enviaría en email de confirmación, con caducidad de 48 horas, para confirmar el registro del usuario.

6.1.4 Mejorar la interfaz gráfica y adaptarla a todos los dispositivos del mercado

Este objetivo o mejora consiste en lograr que la interfaz gráfica sea lo más atractiva posible, aplicando metodologías de diseño centrado en el usuario

o de experiencia del usuario (User Experience), porque lo importante es que una aplicación es para los usuarios, ellos son nuestros verdaderos clientes.

Por otro lado, se considera adecuado que la aplicación web se visualice correctamente tanto desde navegadores web de PC o portátiles, como desde dispositivos móviles (iOS, Android,…).

6.2. Manuales de usuario

Se desarrollaría una funcionalidad de la aplicación, que en cada perfil permitiese consultar el manual de usuario de la aplicación, de tal forma que estuviese disponible online.

6.3. Evaluación de la calidad del código

En una gestión de una implementación real del proyecto se incluiría en la planificación del mismo, desde que se dispusiese de un producto entregable (aunque no completo) la utilización de herramientas de evaluación de la calidad del código, como Sonarqube.

6.4. Pruebas automatizadas

En una gestión de una implementación real del proyecto se incluiría en la planificación del mismo, desde que se dispusiese de un producto entregable (aunque no completo) la aplicación de herramientas de pruebas automatizadas, permitiendo detectar fallos sin necesidad de que un usuario final realizase pruebas con la aplicación.

Este objetivo no implica que no se realicen pruebas unitarias o de integración por parte de usuarios finales, sólo es una herramienta de ayuda adicional a estas pruebas con usuarios finales.

6.4. Seguridad de la aplicación

La aplicación debe ser segura, ya que como la mayoría hoy en día manejan información sensible de los usuarios (unas más que otras), por lo que la información debe viajar cifrada por Internet, se deben verificar los datos introducidos, para evitar ataques como inyección SQL, entre otros posibles.

Los servidores que alojen la aplicación y la base de datos deben estar protegidos, y pueden comunicarse por un túnel VPN. Se deben mantener actualizados y protegidos con software de seguridad, como Antivirus y cortafuegos.

Se debe robustecer la configuración del servidor web, como *Apache*, que es el que recibe todas las peticiones, y también ataques, desde Internet, así como el servidor de aplicaciones, como *Wildfly*, para evitar agujeros de seguridad. Ambos deben mantenerse actualizados. Por supuesto, se debe implementar el protocolo https.

7. Conclusiones

Llevo años trabajando como administrador de sistemas, y posteriormente como programador, porque siempre he considerado que especialmente en el campo de la informática, la experiencia laboral es muy importante, y que junto con unos estudios adecuados proporcionan una visión profesional y académica muy buena.

Mi trayectoria en la UOC me ha proporcionado conocimientos a lo largo de todas las asignaturas cursadas, que he disfrutado aprendiendo, peleándome, a veces desesperado por comprender los enunciados de las PEC o los materiales, pero siempre aprendiendo tecnologías, conocimientos, técnicas de estudio y, técnicas de desarrollo de aplicaciones, todo ello actualizado y al día.

La culminación de dichos estudios es el TFG, y considero que me ha proporcionado una visión en retrospectiva de todos los conocimientos adquiridos, ensamblándolos organizadamente para comprender que cada conocimiento tiene su lugar en el desarrollo de software. Mi opinión es que éste es el verdadero objetivo del TFG, comprender cómo todos los conocimientos adquiridos a lo largo de los estudios se integran en el desarrollo de una aplicación.

Considero que el TFG especialmente ensambla los conocimientos de las siguientes asignaturas, aunque los conocimientos que proporcionan el resto también son necesarios:

- Gestión de proyectos: estos conocimientos se aplican en la planificación inicial de la asignatura, y en su gestión a lo largo de todo el proyecto respondiendo con medidas a los riesgos que se detecten y evaluando el estado del proyecto cada poco tiempo (cada sprint de la metodología Scrum).
- Análisis y diseño con patrones: estos conocimientos se aplican en la etapa de diseño, analizando qué patrones se podrían aplicar en el diseño de la aplicación.
- Diseño y arquitectura de bases de datos: los conocimientos adquiridos en estas asignaturas se aplican durante la fase de diseño de los modelos conceptual y relacional de la base de datos.
- Ingeniería de requisitos: los conocimientos proporcionados por esta asignatura se aplican en la fase de diseño, obteniendo requisitos y documentándolos mediante casos de uso y diagramas de secuencia.
- Diseño y programación orientada a objetos: por supuesto, estos conocimientos se aplican en este TFG, ya que J2EE se basa en la programación orientada a objetos.
- Ingeniería del software de componentes y sistemas distribuidos: los conocimientos de esta asignatura sin duda se aplican en este

TFG puesto que la tecnología J2EE está orientada a entornos distribuidos.

Me decidí por el TFG en la tecnología J2EE porque considero que actualmente tiene mucho auge con Internet y la ubicuidad de las redes, con la aparición y fortalecimiento de las redes distribuidas, así como proporcionando un entorno de trabajo estándar para desarrollar aplicaciones de entorno empresarial.

El objetivo de la planificación temporal del proyecto era desarrollar las funcionalidades que permitiesen comprender el objetivo de la aplicación, se considera que con los perfiles Gestor Libros y Lector se obtiene una idea clara de qué se pretende obtener con la aplicación web, ya que el perfil Vendedor es similar al perfil Gestor Libros pero más limitado en cuanto a funcionalidades. El perfil Administrador se ha definido por si los desarrolladores de la aplicación o administradores de sistemas requieren un usuario en la aplicación, para una gestión mínima.

Respecto a la planificación inicial, estaba previsto implementar los perfiles Lector, Gestor Libros y Vendedor, pero el análisis de la integración de J2EE con Spring causó desvíos en la planificación, por lo que el perfil Vendedor no se pudo implementar, sin embargo no se considera grave ya que el perfil Vendedor contiene un subconjunto de las funcionalidades del perfil Gestor Libros, permitiendo de este modo comprender el funcionamiento principal de la aplicación con los perfiles implementados: Lector y Gestor Libros.

Respecto a la tecnología a implementar, en una planificación inicial, en la capa de negocio se decidió aplicar el *framework* Spring, pero tras un análisis intenso del *framework* y varias pruebas realizadas en un proyecto sencillo, la integración con J2EE no resultaba muy adecuada ni sencilla de implantar, ya que Spring proporciona herramientas para la ejecución de consultas contra la base de datos en el ámbito de transacciones, gestión de *beans*, etc., aspectos que J2EE incluye por defecto. Estas averiguaciones causaron que se tomase la decisión de no aplicar Spring en el proyecto.

El seguimiento de la planificación en las dos primeras entregas (PEC 1 y PEC 2) se ha cumplido sin desviaciones, y el diagrama de planificación temporal se ha considerado muy útil. A la par que esta planificación se procuraba profundizar en la curva de aprendizaje de las tecnologías necesarias para implementar la aplicación web.

En cuanto a J2EE (JSF, Managed Bean y JPA), habiendo cursado los dos semestres anteriores Ingeniería del software de componentes y sistemas distribuidos, y Proyecto de Desarrollo de Software, se disponía de los conocimientos necesarios. Se procuró profundizar en el *framework PrimeFaces*, que se consiguió durante la PEC 2, y en Spring, que conllevó un análisis y un estudio superior a los previstos, causando un retraso en la planificación de la PEC 3.

Respecto a los cambios necesarios para garantizar el éxito del trabajo, se ha tomado la decisión de implementar los perfiles Gestor Libros y Lector, ya que estos dos perfiles permiten conocer las funcionalidades que forman el núcleo de la aplicación web.

Considerando a las metodologías a aplicar a lo largo del desarrollo del producto, considerando las previstas inicialmente, se realiza un análisis:

- Gestión de proyectos: se ha aplicado organizando una planificación hasta la última entrega, y gestionando los riesgos, que principalmente ha sido el estudio de la integración de J2EE con Spring, tomando la decisión de dedicar más tiempo al estudio de Spring, a cambio de sacrificar la implementación del perfil Vendedor, sin dificultar la implementación de las funcionalidades principales que permitiesen comprender el objetivo de la aplicación.
- Metodología ágil: SCRUM. Esta metodología se ha aplicado para evitar el desarrollo en cascada, disponiendo siempre de un producto entregable y controlando los riesgos a tiempo, ya que se basa en Sprint de duración corta.
- *Test driven development*: esta técnica se ha aplicado ya que pruebas extraídas a partir de los casos de uso se han seguido para implementar la aplicación web, logrando que se satisfagan los requisitos.
- Análisis y aplicación de patrones de diseño: Se han analizado los patrones de diseño que se estudiaron en la asignatura correspondiente (Análisis y diseño con patrones), por si alguno fuese aplicable, detectando el patrón MVC (implementado con la tecnología JSF: vistas, el controlador implícito *FacesServlet* y los modelos – *managed bean*-), y el patrón Fachada en la capa de integración.

En cuanto a las líneas de trabajo y posibles mejoras, considerando las mencionadas en el apartado 6 de esta memoria, no se han podido explorar ninguna, son mejoras posibles para una futura ampliación del proyecto.

8. Bibliografía

Henry H. Liu (2015). Spring 4 for Developing Enterprise Applications (An End-to-End Approach), Perfmath.

https://www.primefaces.org/ (fechas de consulta: desde Marzo hasta Junio de 2017)

https://www.primefaces.org/docs/guide/primefaces_user_guide_6_1.pdf (fechas de consulta: desde Marzo hasta Junio de 2017)

https://www.primefaces.org/showcase/ (fechas de consulta: desde Marzo hasta Junio de 2017)

https://docs.oracle.com/javaee/7/tutorial/index.html (fechas de consulta: desde Marzo hasta Junio de 2017)

https://docs.oracle.com/javaee/7/api/toc.htm (fechas de consulta: desde Marzo hasta Junio de 2017)

https://docs.oracle.com/javaee/7/javaserver-faces-2-2/vdldocs-facelets/toc.htm (fechas de consulta: desde Marzo hasta Junio de 2017)

https://en.wikipedia.org/wiki/Java_EE_version_history#Java_EE_7_.28June_12.2C_2013.29 (fecha de consulta: 03/04/2017)

9. Glosario

Término	Descripción
J2EE	Java Platform Enterprise Edition
JSF	Java Server Faces: tecnología del entorno J2EE para la capa de presentación, compuesta por las vistas (documentos xhtml), el controlador Faces Servlet y los Managed Bean (modelo).
Faces Servlet	Controlador implícito, y transparente al programador, del patrón MVC aplicado en la tecnología JSF
Managed Bean	Componente de JSF que forma el modelo del patrón MVC, recogiendo la información introducida por el usuario en las vistas, y remitiéndola a los EJB.
EJB	Enterprise Java Bean
JPA	Java Persistence Api
JPQL	Java Persistence Query Language
PrimeFaces	*Framework* que ofrece funcionalidades adicionales a los elementos existentes en JSF, así como elementos nuevos para las vistas.
Spring Framework	

10. Anexos

Manual de instalación del entorno de ejecución de la aplicación

Manual de los usuarios de perfil Gestor Libros y Lector