

TFC J2EE: Proyecto Biblioteca Virtual

Mariano Ostrovsky
ETIG

Salvador Campo Mazarico

17 de Junio de 2010

A mis padres, por haber creído siempre en mí.

A Anna, por su apoyo y paciencia infinita.

A Simó, por hacerme tan feliz.

Resumen

Como Trabajo de Fin de Carrera (TFC) se ha realizado el análisis, diseño e implementación de una aplicación utilizando tecnología Java y una arquitectura J2EE. El presente documento contiene la memoria del trabajo realizado durante el desarrollo de una Biblioteca Virtual.

El proyecto Biblioteca Virtual consiste en una aplicación accesible a través de Internet, en la que los usuarios pueden consultar el catálogo de libros disponibles, solicitarlos, realizar búsquedas específicas, etc. Además, presenta opciones para administrar el stock de libros y los préstamos.

Para llevar a cabo el desarrollo se han utilizado los conocimientos adquiridos a lo largo de la carrera de Ingeniería Técnica de Informática de Gestión en programación, análisis orientado a objetos y diseño de bases de datos. A su vez, ha sido necesario el estudio de la tecnología J2EE y de los diversos frameworks y componentes disponibles en el mercado.

Área del TFC: J2EE

Palabras clave: Java, J2EE, JavaServer Faces, Hibernate, Patrones de diseño.

Licencia: Esta trabajo está sujeto – salvo que se indique lo contrario – a una licencia de Reconocimiento-No comercial-Sin obras derivadas 2.5 España de Creative Commons. Podéis copiarlo, distribuirlo i transmitirlo públicamente siempre que citéis el autor y la obra, no se haga un uso comercial y no se haga copia derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.

Índice de contenidos

Resumen.....	3
Índice de contenidos	4
Índice de figuras	6
1. Introducción	8
1.1. Justificación del TFC: punto de partida y aportación	8
1.2. Objetivos del TFC.....	8
1.3. Enfoque y método seguido	9
1.4. Planificación del proyecto	9
1.4.1. Calendario	9
1.4.2. Riesgos.....	10
1.4.3. Tareas a desarrollar.....	10
1.4.4. Planificación temporal.....	11
1.5. Productos obtenidos	11
1.6. Descripción de los capítulos de la memoria.....	11
2. Descripción del proyecto Biblioteca Virtual	13
3. Análisis.....	14
3.1. Actores	14
3.2. Casos de uso	14
3.3. Descripción textual de los casos de uso	15
3.4. Prototipo	20
4. Diseño.....	27
4.1. Diagrama de clases.....	27
4.2. Diagramas de estado.....	28
4.3. Diagramas de actividad	29
4.4. Diagramas de secuencia	31
4.5. Diseño de la base de datos.....	34
4.6. Arquitectura de la aplicación	35
5. Implementación	37
5.1. Decisiones de diseño e implementación.....	37
5.2. Diagrama de navegación	38
5.3. Diagrama de packages	38
5.4. Requerimientos de software.....	39

5.5. Script de creación de la Base de Datos	40
5.6. Instalación de la aplicación	42
6. Conclusiones.....	45
Glosario	46
Bibliografía	47
Anexos.....	49
Imágenes del sitio web Biblioteca Virtual	49

Índice de figuras

Ilustración 1. Planificación temporal del proyecto	11
Ilustración 2. Diagrama de casos de uso	14
Ilustración 3. Prototipo - Registro de nuevo socio	20
Ilustración 4. Prototipo - Conexión al sistema	21
Ilustración 5. Prototipo - Búsqueda por categoría	21
Ilustración 6. Prototipo - Libros más valorados	22
Ilustración 7. Prototipo - Ver detalle de libro	22
Ilustración 8. Prototipo - Ver mis libros	23
Ilustración 9. Prototipo - Entregar libro	23
Ilustración 10. Prototipo - Retornar libro.....	24
Ilustración 11. Prototipo - Gestionar catálogo.....	24
Ilustración 12. Prototipo - Modificar libro	25
Ilustración 13. Prototipo - Ver solicitudes.....	25
Ilustración 14. Prototipo - Desbloquear usuario.....	26
Ilustración 15. Diagrama de clases.....	27
Ilustración 16. Diagrama de estado - Estados de un socio.....	28
Ilustración 17. Diagrama de estado - Tipos de préstamo	28
Ilustración 18. Diagrama de estado - Estados del préstamo.....	29
Ilustración 19. Diagrama de estado - Estados de un ejemplar.....	29
Ilustración 20. Diagrama de actividad - Solicitud de préstamo	29
Ilustración 21. Diagrama de actividad - Entrega de ejemplar	30
Ilustración 22. Diagrama de actividad - Renovación y devolución de un ejemplar	30
Ilustración 23. Diagrama de secuencia - Identificación en el sistema	31
Ilustración 24. Diagrama de secuencia - Búsqueda de libros.....	31
Ilustración 25. Diagrama de secuencia - Registro de un nuevo socio.....	32
Ilustración 26. Diagrama de secuencia - Ver libro.....	32
Ilustración 27. Diagrama de secuencia - Valorar libro	33
Ilustración 28. Diagrama de secuencia - Solicitar préstamo	33
Ilustración 29. Diagrama de secuencia - Entregar libro	34
Ilustración 30. Diagrama de la base de datos	34
Ilustración 31. Arquitectura de la aplicación	35
Ilustración 32. Diagrama de navegación	38
Ilustración 33. Diagrama de packages.....	38
Ilustración 34. Desplegar WAR.....	42
Ilustración 35. Estructura de directorios.....	42
Ilustración 36. Biblioteca Virtual - Registro de un nuevo socio	49
Ilustración 37. Biblioteca Virtual - Conexión al sistema.....	50
Ilustración 38. Biblioteca Virtual - Búsqueda de últimas novedades.....	50
Ilustración 39. Biblioteca Virtual - Ver libro	51
Ilustración 40. Biblioteca Virtual - Elección de fecha al solicitar el libro.....	51
Ilustración 41. Biblioteca Virtual - Ver mis libros.....	52

Ilustración 42. Biblioteca Virtual - Ver solicitudes	52
Ilustración 43. Biblioteca Virtual - Entregar libro.....	53
Ilustración 44. Biblioteca Virtual - Retornar libro	53
Ilustración 45. Biblioteca Virtual - Gestionar categorías.....	54
Ilustración 46. Biblioteca Virtual - Gestionar libros	54
Ilustración 47. Biblioteca Virtual - Modificar datos del libro	55
Ilustración 48. Biblioteca Virtual - Modificar autores del libro	55
Ilustración 49. Biblioteca Virtual - Modificar categorías del libro.....	56
Ilustración 50. Biblioteca Virtual - Modificar ejemplares del libro	56
Ilustración 51. Biblioteca Virtual - Desbloquear socio	57

1. Introducción

Como Trabajo de Fin de Carrera (TFC) se realizará el análisis, diseño e implementación de una aplicación utilizando tecnología Java y una arquitectura J2EE. El TFC consistirá en el desarrollo de una Biblioteca Virtual, accesible a través de Internet, en la que los usuarios podrán consultar el catálogo de libros disponibles, solicitarlos, realizar búsquedas específicas, etc.

1.1. Justificación del TFC: punto de partida y aportación

El TFC es una asignatura que está pensada para realizar un trabajo de síntesis de los conocimientos adquiridos en otras asignaturas de la carrera y que requiere ponerlos en práctica conjuntamente en un proyecto concreto. Normalmente, y como es el caso, el TFC es un trabajo eminentemente práctico y vinculado al ejercicio profesional de la informática.

La elección de esta área de trabajo ha estado motivada por varios factores. Por un lado, me ha permitido experimentar el desarrollo de un proyecto desde cero utilizando una metodología rigurosa y académica. Por otra parte, y aunque desde adolescente he estado ligado al mundo de la programación (primero de forma amateur y luego de forma profesional), me ha servido para acercarme por primera vez al mundo de J2EE y la programación web.

Una vez elegida el área, he buscado un proyecto que pudiese ser desarrollado completamente en los tiempos previstos para la realización del TFC, que fuese similar a los que nos podemos encontrar en una empresa y que permitiese investigar diferentes opciones de implementación dentro la arquitectura J2EE. Es en este contexto donde surge la idea de realizar una aplicación que implemente la gestión de una biblioteca a través de la web.

La principal aportación del proyecto Biblioteca Virtual es brindar una aplicación sencilla a las bibliotecas de barrio que por razones presupuestarias no pueden costear el desarrollo de un sistema a medida. Esta aplicación ofrece los servicios indispensables para la consulta del catálogo y la gestión de préstamos.

1.2. Objetivos del TFC

El objetivo principal de este TFC es profundizar en el uso de la tecnología Java y adquirir conocimientos sobre la arquitectura J2EE mediante el desarrollo del proyecto Biblioteca Virtual. Con este fin, se trabaja en las siguientes áreas:

- Análisis, diseño e implementación de una aplicación similar a las existentes en el mundo empresarial utilizando la orientación a objetos.
- Patrones de diseño aplicables a la arquitectura J2EE.
- Comunicación con bases de datos relacionales.
- Tecnología J2EE: JavaServer Pages (JSP), Servlets, Enterprise Java Beans (EJB), etc.

- Frameworks: Hibernate, JavaServer Faces, etc.

Por otra parte, al pretender ser este un trabajo de síntesis de los conocimientos adquiridos en la carrera, se siguen las metodologías de desarrollo y programación ya estudiadas en otras asignaturas. Al final del TFC la aplicación debe ser completamente funcional.

1.3. Enfoque y método seguido

Para el desarrollo de la aplicación se ha intentado seguir el enfoque de una empresa de desarrollo de software. Esto implica que no sólo es importante el resultado final, sino que éste debe haberse acotado previamente de manera clara, debe generarse la documentación adecuada a cada fase y deben alcanzarse la consecución de los objetivos en los plazos establecidos.

Para ello se ha utilizado un ciclo en cascada clásico en el que se han previsto y realizado las siguientes etapas:

- Definición funcional
- Planificación del proyecto
- Análisis
- Diseño
- Implementación
- Pruebas
- Documentación

En paralelo, desde el comienzo del TFC se ha avanzado en el estudio de la arquitectura J2EE y del software necesario para su ejecución. En primer lugar, se ha realizado una breve investigación de las opciones disponibles. Una vez vistas las opciones más adecuadas al proyecto se han realizado algunas pruebas de implementación para evaluarlas. Finalmente, definidas las principales tecnologías a utilizar, en concreto JavaServer Faces e Hibernate, se ha buscado material bibliográfico que permitiese la formación en dichas tecnologías y sirviese de referencia durante la implementación del proyecto.

1.4. Planificación del proyecto

Teniendo en cuenta la descripción del proyecto Biblioteca Virtual y los objetivos a conseguir se procede a establecer una planificación que nos permita realizar con éxito el TFC.

1.4.1. Calendario

Para la realización del TFC existen las siguientes fechas límite marcadas por el consultor:

- Plan de trabajo: 10/03/2010.
- PAC 2: 15/04/2010.
- PAC 3: 24/05/2010.
- Memoria y presentación virtual: 17/06/2010.

Para cumplir con estas fechas se ha previsto trabajar de lunes a domingo, ajustando la dedicación a los compromisos laborales existentes entre semana.

1.4.2. Riesgos

Se detectan los siguientes riesgos que pueden provocar retrasos en la planificación:

- Falta de experiencia en el entorno J2EE.
- Compromisos laborales imprevistos.
- Compromisos familiares.

1.4.3. Tareas a desarrollar

Una vez analizada la descripción del proyecto, podemos asumir que los requisitos no cambiarán durante la realización del TFC. Usando un ciclo de vida en cascada clásico podemos prever las siguientes tareas a desarrollar:

Documento a entregar	Fase	Tarea	Duración estimada
Plan de trabajo	Plan de trabajo	Definición de los requerimientos	2 días
		Definición de objetivos	1 día
		Planificación del proyecto	1 día
	Análisis	Descripción de casos de uso e identificación de actores	4 días
		Prototipo	3 días
		Glosario	1 día
PAC 2	Diseño	Revisión del análisis	3 días
		Diagramas de clases	5 días
		Diagramas de secuencia, estado y actividad más destacados	6 días
		Diseño de la base de datos	4 días
		Diseño de la arquitectura	11 días
PAC 3	Implementación	Instalación del entorno de trabajo	5 días
		Creación de la base de datos	3 días
		Programación	23 días
		Pruebas	5 días
		Manual de instalación	2 días
Memoria y Presentación Virtual	Memoria y Presentación Virtual	Repaso del proyecto realizado	2 días
		Correcciones	2 días
		Redacción de la memoria	12 días
		Realización de la presentación	7 días

1.4.4. Planificación temporal

Teniendo en cuenta las restricciones temporales dadas y las tareas a realizar, obtenemos el siguiente diagrama de Gantt:

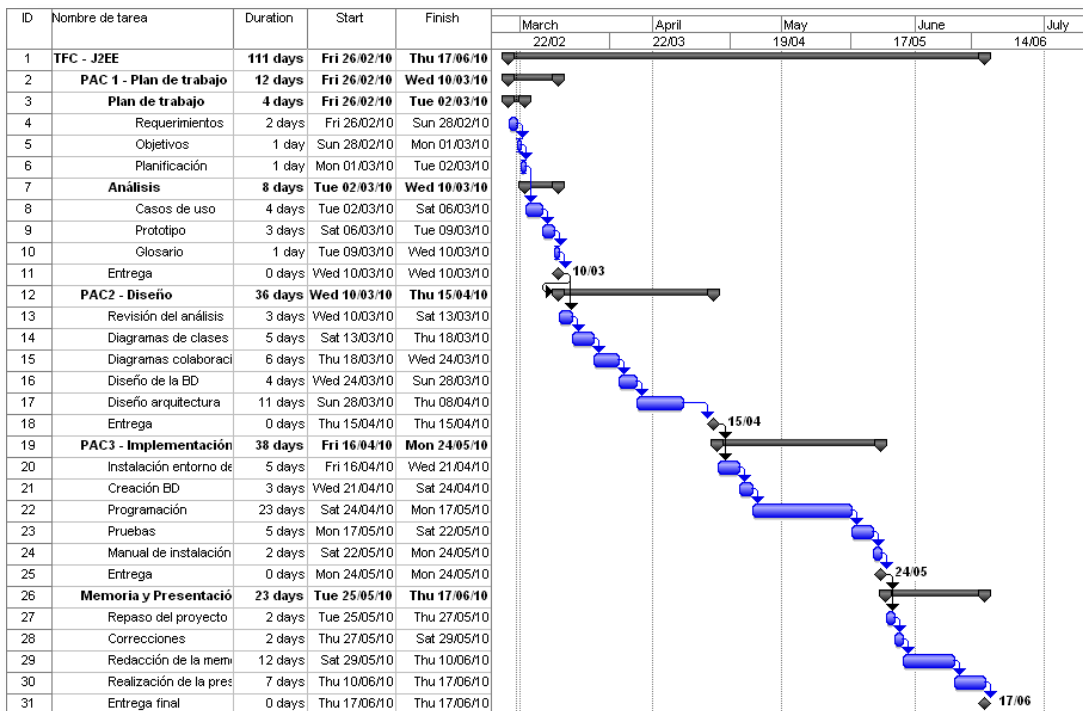


Ilustración 1. Planificación temporal del proyecto

1.5. Productos obtenidos

Los productos obtenidos en la realización de este TFC consisten en:

- Una aplicación web completamente funcional que permite la gestión de una biblioteca a través de Internet.
- La presente memoria, con la síntesis del trabajo realizado.
- La presentación, que resume el trabajo realizado y los resultados obtenidos.

1.6. Descripción de los capítulos de la memoria

En el resto de capítulos de la memoria se presentan los siguientes aspectos:

- **Descripción del proyecto Biblioteca Virtual:** especifica y delimita el alcance funcional esperado de la aplicación a desarrollar.
- **Análisis:** se identifican los actores y casos de uso principales utilizando el análisis orientado a objetos.
- **Diseño:** se identifican las clases de entidad principales, se presentan los diagramas de estado, actividad y secuencia más destacados, el diseño de la base de datos y finalmente se define la arquitectura de la aplicación web.

- **Implementación:** se presentan las decisiones de diseño e implementación tomadas durante el desarrollo. También se indican los requerimientos de software, así como los pasos a realizar para una correcta configuración y ejecución de la aplicación.
- **Conclusiones:** se presentan las principales conclusiones obtenidas una vez finalizada la implementación.

2. Descripción del proyecto Biblioteca Virtual

El proyecto Biblioteca Virtual está dirigido a bibliotecas de barrio que estén interesadas en ofrecer sus servicios a través de internet. Tiene por objetivo facilitar la consulta del catálogo disponible así como permitir la solicitud de ejemplares y renovación de préstamos mediante una interfaz de usuario sencilla y amigable.

Al acceder a la web de la biblioteca, los usuarios (sin importar si se han identificado) podrán consultar los últimos libros agregados y los libros más recomendados por otros usuarios. También podrán realizar búsquedas de libros y ver el detalle de un libro. Al seleccionar ver el detalle de un libro se verá: una foto de la portada, el título, los autores, la fecha de publicación, el ISBN, la editorial, las categorías a las que pertenece, la descripción del mismo y los comentarios y puntuaciones de los usuarios.

Los usuarios podrán convertirse en socios de la biblioteca completando un formulario de registro disponible en la web. Una vez realizado este paso podrán, previa identificación, solicitar un libro para una fecha concreta por un plazo de treinta días y realizar hasta tres renovaciones del préstamo siempre y cuando el ejemplar en poder del socio no esté pedido por otro usuario.

Cada vez que se realice una solicitud o renovación se enviará un correo electrónico de aviso al socio y a una cuenta de la biblioteca habilitada especialmente. Las renovaciones no podrán realizarse hasta cinco días antes de que acabe el préstamo. A su vez, cada socio no podrá tener en su poder más de tres libros simultáneamente.

El socio, además, podrá consultar los libros que tiene en préstamo en la actualidad y las solicitudes pendientes. Si fuese el caso, podrá cancelar una solicitud que aún no ha sido satisfecha. Una funcionalidad muy importante del sistema es la que permite votar y opinar sobre los libros que el socio ya ha leído independientemente de que haya obtenido el libro de la biblioteca.

La Biblioteca Virtual tendrá al menos un administrador que se encargará de gestionar el catálogo y el stock disponible de cada libro, informar cuando un ejemplar es entregado a un socio y cuando es devuelto. Cuando un libro se devuelva fuera de plazo, el sistema penalizará al socio impidiéndole volver a solicitar préstamos hasta que el administrador lo autorice.

El administrador también podrá consultar las solicitudes pendientes, los préstamos en curso y rechazar una solicitud de préstamo.

3. Análisis

En este apartado avanzaremos en el análisis del problema propuesto, describiendo los actores y los casos de uso más importantes. También se presentará un prototipo del sitio web a desarrollar y un glosario.

3.1. Actores

Los actores que interactúan con la Biblioteca Virtual son:

- **Invitado:** Son aquellos usuarios que consultan el catálogo sin identificarse previamente. Pueden consultar el material disponible pero no pueden solicitar préstamos.
- **Socio:** Los socios al igual que los invitados pueden consultar el catálogo de libros. Pero a diferencia de estos, están habilitados, previa identificación, para solicitar préstamos de libros, renovarlos y/o cancelar solicitudes aún no satisfechas.
- **Administrador:** Son usuarios con permisos especiales que les permiten gestionar el catálogo de libros, dando de alta nuevos ejemplares, administrar su stock, confirmar la entrega y/o devolución de un libro, etc.

3.2. Casos de uso

De la descripción del proyecto Biblioteca Virtual podemos identificar los siguientes casos de uso:

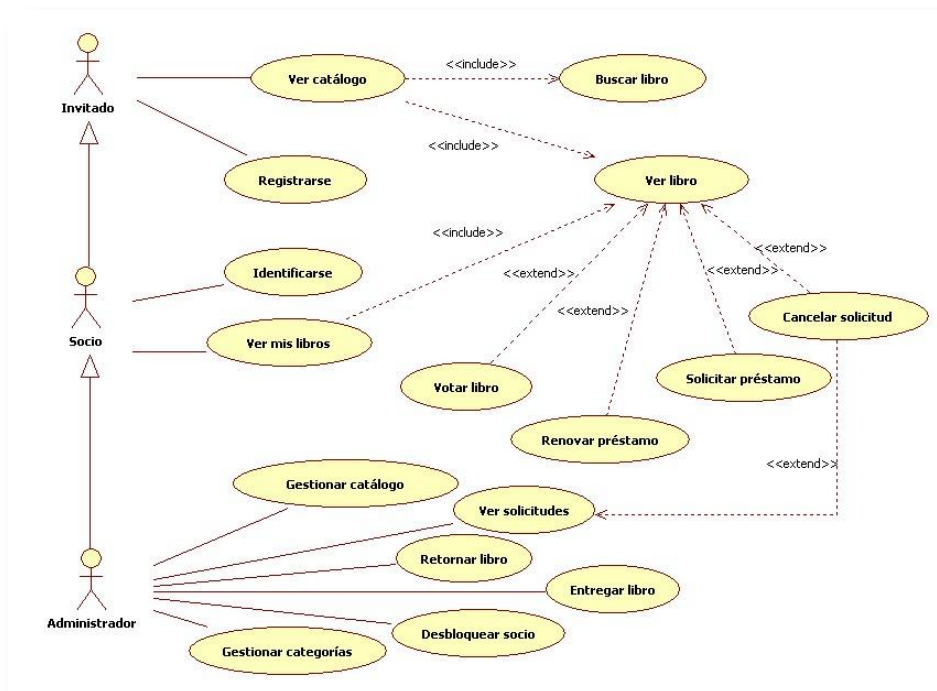


Ilustración 2. Diagrama de casos de uso

3.3. Descripción textual de los casos de uso

1. Caso de uso **Ver catálogo**
 - 1.1. Actor principal: **Invitado**
 - 1.2. Precondición: Ninguna
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Buscar libro, Ver libro**
2. Escenario principal:
 - 2.1. El usuario se conecta a la Biblioteca Virtual.
 - 2.2. El sistema muestra los últimos libros incorporados al catálogo.
 - 2.3. El usuario solicita ver los libros más recomendados del catálogo.
 - 2.4. El sistema muestra los libros del catálogo ordenados por valoración.
 - 2.5. El usuario solicita realizar la búsqueda de un libro en concreto.
 - 2.6. El sistema ejecuta el caso de uso **Buscar Libro**.
3. Flujos alternativos:
 - 3.1. En cualquier momento el usuario selecciona un libro para ver su detalle.
 - 3.2. El sistema ejecuta el caso de uso **Ver Libro**.

1. Caso de uso **Buscar libro**
 - 1.1. Actor principal: **Invitado**
 - 1.2. Precondición: Ninguna
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Ver Catálogo**
2. Escenario principal:
 - 2.1. El usuario selecciona un criterio de búsqueda (Título, ISBN, Categoría, Autor) e ingresa un texto.
 - 2.2. El sistema devuelve aquellos libros que contienen el texto dentro del criterio seleccionado.
3. Flujos alternativos:
 - 3.1. Si no hay libros que cumplan los criterios de búsqueda el sistema muestra un mensaje al usuario.

1. Caso de uso **Ver libro**
 - 1.1. Actor principal: **Invitado, Socio**
 - 1.2. Precondición: Un libro existente en el catálogo
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Ver catálogo, Ver mis libros, Votar libro, Solicitar préstamo, Renovar préstamo, Cancelar solicitud, Identificarse**
2. Escenario principal:
 - 2.1. El usuario indica el libro del que quiere ver el detalle.
 - 2.2. El sistema muestra al usuario la información asociada al libro: Título, descripción, ISBN, fecha de publicación, categorías, autores y una foto de la portada. También muestra la valoración promedio del libro y los últimos comentarios de los usuarios que lo han valorado.
3. Flujos alternativos:
 - 3.1. Si el usuario es socio, se ha identificado previamente y aún no ha valorado el libro, el sistema le da la opción de valorarlo.
 - 3.2. El sistema ejecuta el caso de uso **Votar libro**.
 - 3.3. Si el usuario es socio, se ha identificado previamente y no tiene el libro actualmente en préstamo, el sistema le da la opción de solicitarlo.

- 3.4. El sistema ejecuta el caso de uso **Solicitar préstamo**.
- 3.5. Si el usuario es socio, se ha identificado previamente y tiene el libro en su poder, el sistema le da la opción de renovar el préstamo.
- 3.6. El sistema ejecuta el caso de uso **Renovar préstamo**.
- 3.7. Si el usuario es socio, se ha identificado previamente, ha solicitado el libro y aún no se le ha entregado, el sistema le da la opción de cancelar su solicitud.
- 3.8. El sistema ejecuta el caso de uso **Cancelar solicitud**.

- 1. Caso de uso **Registrarse**
 - 1.1. Actor principal: **Invitado**
 - 1.2. Precondición: Ninguna
 - 1.3. Postcondición: existe un nuevo socio en el sistema
 - 1.4. Casos de uso relacionados: ninguno
- 2. Escenario principal:
 - 2.1. El invitado indica su nombre, apellido, dirección, ciudad, país, dirección de e-mail, código de usuario y contraseña.
 - 2.2. El sistema registra un nuevo socio en el sistema.
 - 2.3. El sistema envía un correo al usuario confirmando el registro como socio.
- 3. Flujos alternativos:
 - 3.1. Si la dirección de e-mail ya es utilizada por otro usuario o el nombre de usuario ya existe muestra un error y permite volver a introducir los datos erróneos.

- 1. Caso de uso **Identificarse**
 - 1.1. Actor principal: **Socio**
 - 1.2. Precondición: Ninguna
 - 1.3. Postcondición: El usuario está identificado
 - 1.4. Casos de uso relacionados: **Ver mis libros, Ver libro, Votar libro, Solicitar préstamo, Renovar préstamo, Cancelar solicitud, Gestionar catálogo, Ver solicitudes, Entregar libro, Retornar libro, Desbloquear socio**.
- 2. Escenario principal:
 - 2.1. El usuario indica su código de usuario y contraseña.
 - 2.2. El sistema verifica que se trata de un usuario válido
- 3. Flujos alternativos:
 - 3.1. Si el usuario no existe en el sistema o la contraseña es incorrecta muestra un error.

- 1. Caso de uso **Ver mis libros**
 - 1.1. Actor principal: **Socio**
 - 1.2. Precondición: El usuario se ha identificado previamente
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Identificarse, Ver libro**
- 2. Escenario principal:
 - 2.1. El usuario solicita ver los libros que ha solicitado.
 - 2.2. El sistema muestra al usuario los libros que el usuario tiene actualmente en préstamo y los solicitados que aún no han sido entregados.
- 3. Flujos alternativos:
 - 3.3. El usuario selecciona un libro para ver su detalle.
 - 3.4. El sistema ejecuta el caso de uso **Ver Libro**.

1. Caso de uso **Votar libro**
 - 1.1. Actor principal: **Socio**
 - 1.2. Precondición: El usuario se ha identificado previamente y no ha valorado previamente el libro
 - 1.3. Postcondición: El libro está valorado por el usuario
 - 1.4. Casos de uso relacionados: **Ver libro**
2. Escenario principal:
 - 2.1. El usuario puede valorar del 1 (muy malo) al 5 (muy bueno) el libro y escribir un pequeño comentario sobre el mismo.
 - 2.2. El sistema registra el comentario del usuario.
3. Flujos alternativos: Ninguno

1. Caso de uso **Solicitar préstamo**
 - 1.1. Actor principal: **Socio**
 - 1.2. Precondición: El usuario se ha identificado previamente y no está bloqueado, no tiene el libro en su poder ni lo ha solicitado previamente. El libro existe en el catálogo y tiene stock positivo.
 - 1.3. Postcondición: El usuario tiene solicitado el libro
 - 1.4. Casos de uso relacionados: **Identificarse, Ver libro**
2. Escenario principal:
 - 2.1. El usuario elige una fecha y solicita un libro.
 - 2.2. El sistema comprueba que en la fecha solicitada, y hasta treinta días después, el libro no se encuentra ni en préstamo ni está solicitado por otro usuario.
 - 2.3. El sistema comprueba que en las fechas de solicitud del ejemplar, el usuario no tendrá en su poder más de tres libros.
 - 2.4. El sistema envía un e-mail al socio y a una cuenta especial de la biblioteca informando de la solicitud de préstamo y del código de ejemplar solicitado.
3. Flujos alternativos:
 - 3.1. Si la fecha es menor a la fecha del día, el sistema muestra un error.
 - 3.2. Si el libro no se puede solicitar en las fechas señaladas, el sistema muestra un aviso y propone la fecha más próxima en la que la solicitud puede ser satisfecha.

1. Caso de uso **Renovar préstamo**
 - 1.1. Actor principal: **Socio**
 - 1.2. Precondición: El usuario se ha identificado previamente y no está bloqueado, tiene el libro en su poder, quedan menos de 5 días para que se acabe el préstamo y no ha renovado más de tres veces el préstamo
 - 1.3. Postcondición: El usuario renueva el préstamo sobre el libro
 - 1.4. Casos de uso relacionados: **Identificarse, Ver libro**
2. Escenario principal:
 - 2.1. El usuario solicita renovar un libro que tiene en su poder.
 - 2.2. El sistema comprueba que en la fecha de finalización del préstamo, y hasta treinta días después, el ejemplar no está solicitado por otro usuario.
 - 2.3. El sistema comprueba que en las fechas de renovación del ejemplar, el usuario no tendrá en su poder más de tres libros.
 - 2.4. El sistema envía un e-mail al socio y a una cuenta especial de la biblioteca informando de la renovación del préstamo.
3. Flujos alternativos:
 - 3.1. Si el libro no se puede renovar, el sistema muestra un error.

1. Caso de uso **Cancelar solicitud**
 - 1.1. Actor principal: **Socio, Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente, ha solicitado el libro y aún no lo tiene en su poder
 - 1.3. Postcondición: El libro deja de estar solicitado
 - 1.4. Casos de uso relacionados: **Identificarse, Ver libro, Ver solicitudes**
2. Escenario principal:
 - 2.1. El usuario solicita cancelar el préstamo sobre el libro.
 - 2.2. El sistema envía un e-mail al socio y a una cuenta especial de la biblioteca informando de la cancelación de la solicitud.
3. Flujos alternativos: Ninguno

1. Caso de uso **Gestionar catálogo**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Identificarse**
2. Escenario principal:
 - 2.1. El sistema solicita el ISBN del libro.
 - 2.2. El usuario lo informa y elige entre crear, modificar o borrar el libro del catálogo.
 - 2.3. Si la opción elegida por el usuario es crear o modificar, el sistema solicita información sobre el libro.
 - 2.4. El usuario informa el título, la descripción, los autores, las categorías a las que pertenece el libro, el año de publicación, la editorial, el idioma, la foto de portada y los ejemplares disponibles.
 - 2.5. Una vez terminado, el usuario confirma los cambios introducidos.
 - 2.6. El sistema registra los cambios.
3. Flujos alternativos:
 - 3.1. Si el usuario intenta eliminar libros del catálogo con ejemplares en préstamo, el sistema muestra un error.
 - 3.2. Si el usuario intenta dar de alta un libro con un código ISBN ya existente el sistema muestra un error.
 - 3.3. Para cada libro en el catálogo, el usuario puede agregar o quitar ejemplares.
 - 3.4. Si el usuario intenta eliminar ejemplares en préstamo, el sistema muestra un error.

1. Caso de uso **Gestionar categorías**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Identificarse**
2. Escenario principal:
 - 2.1. El sistema solicita el nombre de la categoría.
 - 2.2. El usuario lo informa y elige entre crear, modificar o borrar la categoría.
 - 2.3. El usuario confirma la opción deseada.
 - 2.4. El sistema registra los cambios.
3. Flujos alternativos:
 - 3.1. Si el usuario intenta eliminar categorías en uso por algún libro el sistema muestra un error.
 - 3.2. Si el usuario intenta dar de alta una categoría ya existente el sistema muestra un error.

1. Caso de uso **Ver solicitudes**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente
 - 1.3. Postcondición: Ninguna
 - 1.4. Casos de uso relacionados: **Identificarse**,
2. Escenario principal:
 - 2.1. El sistema muestra una lista de las solicitudes, mostrando el nombre del libro, el código de ejemplar, el usuario de la solicitud, la fecha de solicitud, la fecha de comienzo del préstamo y el estado (en préstamo o pendiente).
3. Flujos alternativos:
 - 3.1. El usuario decide cancelar una solicitud pendiente.
 - 3.2. El sistema ejecuta el caso de uso **Cancelar solicitud**.

1. Caso de uso **Entregar libro**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente y no está bloqueado, el ejemplar no está en préstamo
 - 1.3. Postcondición: El ejemplar está en préstamo
 - 1.4. Casos de uso relacionados: **Identificarse**
2. Escenario principal:
 - 2.1. El usuario ingresa el código de socio y el código de ejemplar.
 - 2.2. El sistema verifica que el ejemplar está solicitado por el socio.
 - 2.3. El sistema registra la entrega del libro.
3. Flujos alternativos:
 - 3.1. Si el ejemplar no ha sido solicitado por el socio o se encuentra en préstamo, el sistema muestra un error.

1. Caso de uso **Retornar libro**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El usuario se ha identificado previamente, el ejemplar está en préstamo
 - 1.3. Postcondición: El ejemplar está disponible para ser prestado
 - 1.4. Casos de uso relacionados: **Identificarse**
2. Escenario principal:
 - 2.1. El usuario ingresa el código de socio y el código de ejemplar.
 - 2.2. El sistema verifica que el ejemplar estaba pendiente de retornar y que se realiza la devolución en los plazos previstos.
 - 2.3. El sistema registra la devolución del libro.
3. Flujos alternativos:
 - 3.1. Si el ejemplar no está pendiente de retornar, el sistema muestra un error.
 - 3.2. Si el libro se retorna fuera de plazo, el sistema bloquea al socio.

1. Caso de uso **Desbloquear socio**
 - 1.1. Actor principal: **Administrador**
 - 1.2. Precondición: El socio está bloqueado
 - 1.3. Postcondición: El socio está desbloqueado
 - 1.4. Casos de uso relacionados: **Identificarse**

2. Escenario principal:
 - 2.1. El usuario ingresa el código de socio.
 - 2.2. El sistema verifica que el socio está bloqueado.
 - 2.3. El sistema registra el desbloqueo del socio.
3. Flujos alternativos: ninguno

3.4. Prototipo

Partiendo de la premisa que se desea un sitio web con una interfaz de usuario muy sencilla se presenta un prototipo de las principales pantallas del sistema. El mismo podrá variar en las fases de diseño e implementación incorporando mejoras en la usabilidad y estética.

Este prototipo no pretende ser una presentación exhaustiva de todas las pantallas de la aplicación, sino permitir que en esta fase de la construcción de la Biblioteca Virtual podamos tener una primera aproximación a lo que será el aspecto visual del proyecto una vez finalizado.

Como rasgo distintivo de la aplicación web podemos ver que en la parte de arriba estará siempre disponible la opción de realizar búsquedas. A su vez, en el margen izquierdo estarán las opciones disponibles que cambiarán en función del perfil del usuario conectado. Finalmente, en el margen derecho se mostrarán datos y controles de edición en función de la opción elegida por el usuario.

Registro de nuevo socio:

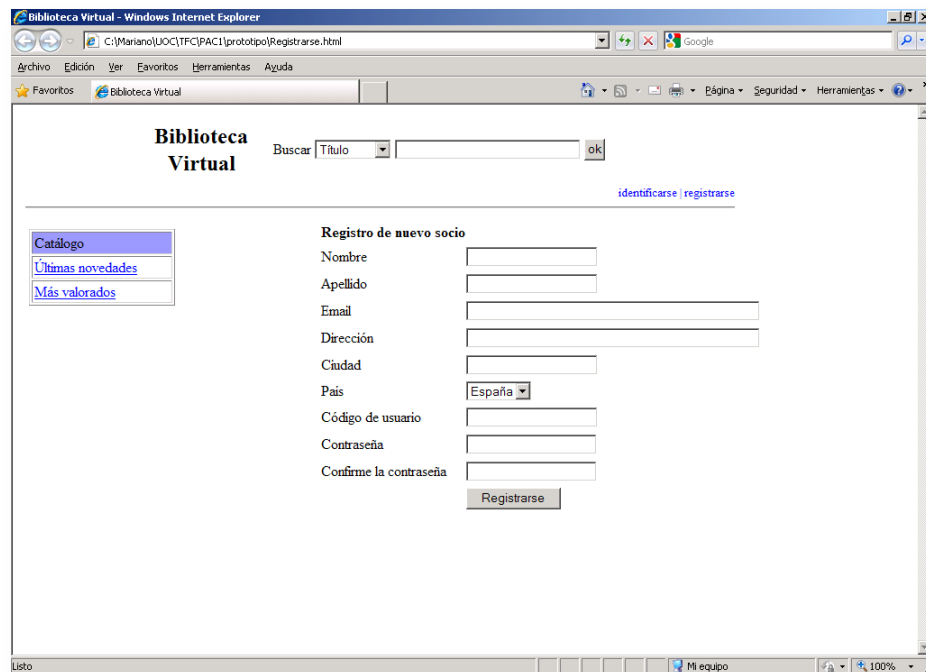


Ilustración 3. Prototipo - Registro de nuevo socio

Conexión al sistema:

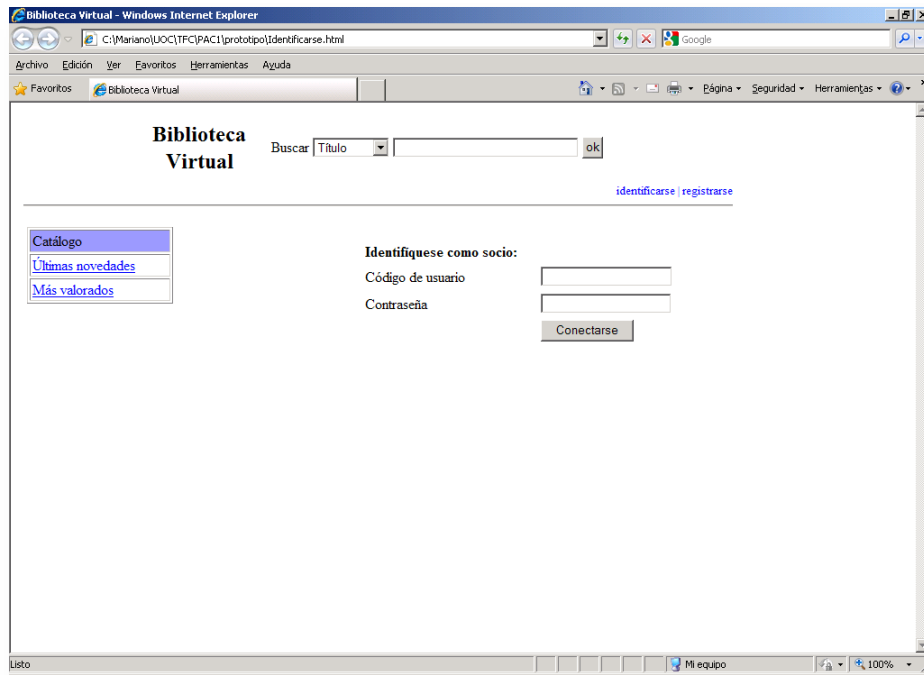


Ilustración 4. Prototipo - Conexión al sistema

Búsqueda por categoría (conectado como socio):

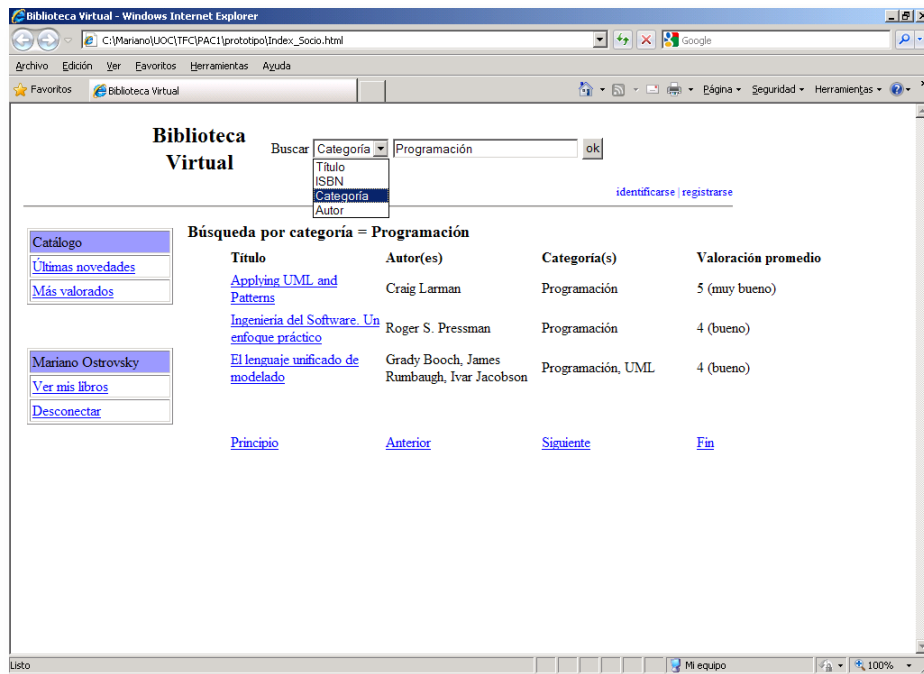


Ilustración 5. Prototipo - Búsqueda por categoría

Libros más valorados (conectado como administrador):

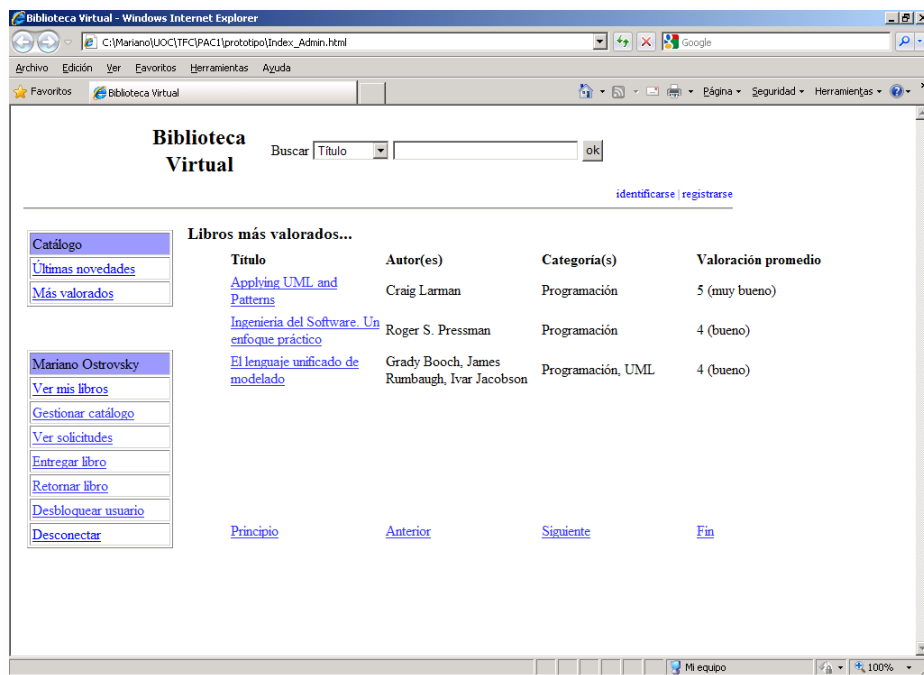


Ilustración 6. Prototipo - Libros más valorados

Ver detalle de libro:

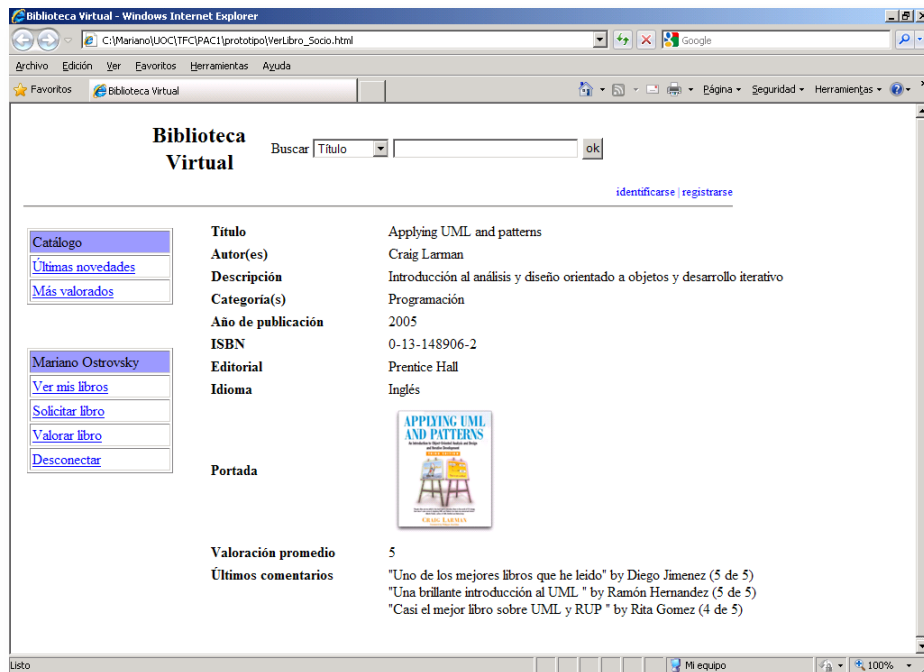


Ilustración 7. Prototipo - Ver detalle de libro

Ver mis libros:

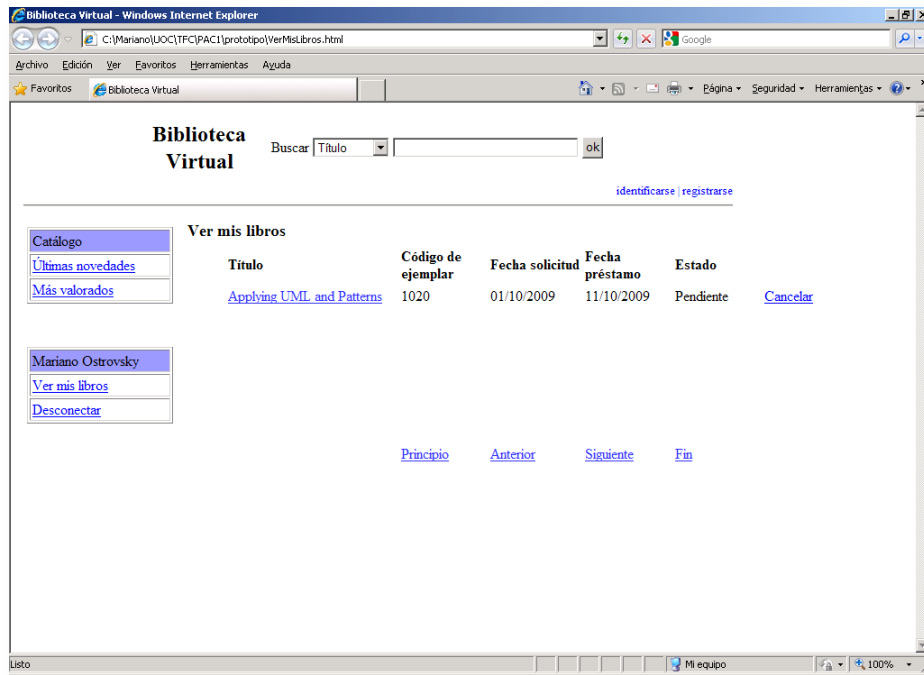


Ilustración 8. Prototipo - Ver mis libros

Entregar libro:

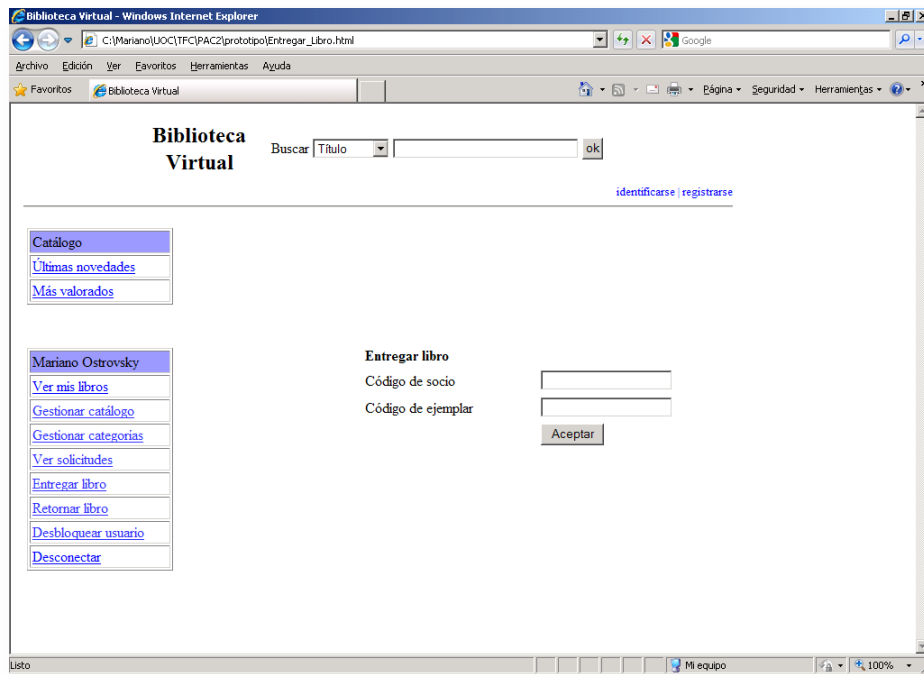


Ilustración 9. Prototipo - Entregar libro

Retornar libro:

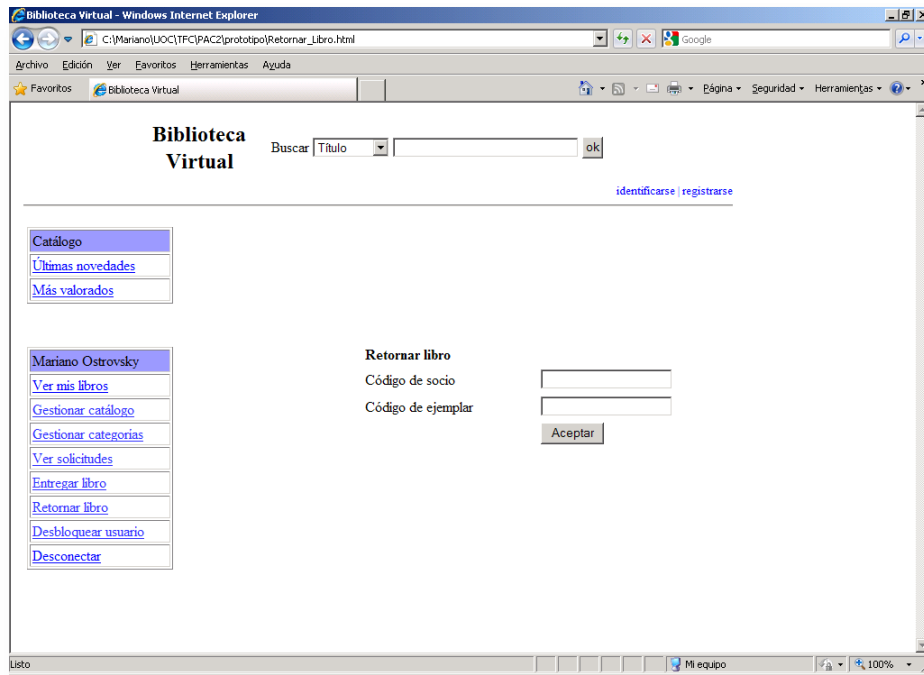


Ilustración 10. Prototipo - Retornar libro

Gestionar catálogo:

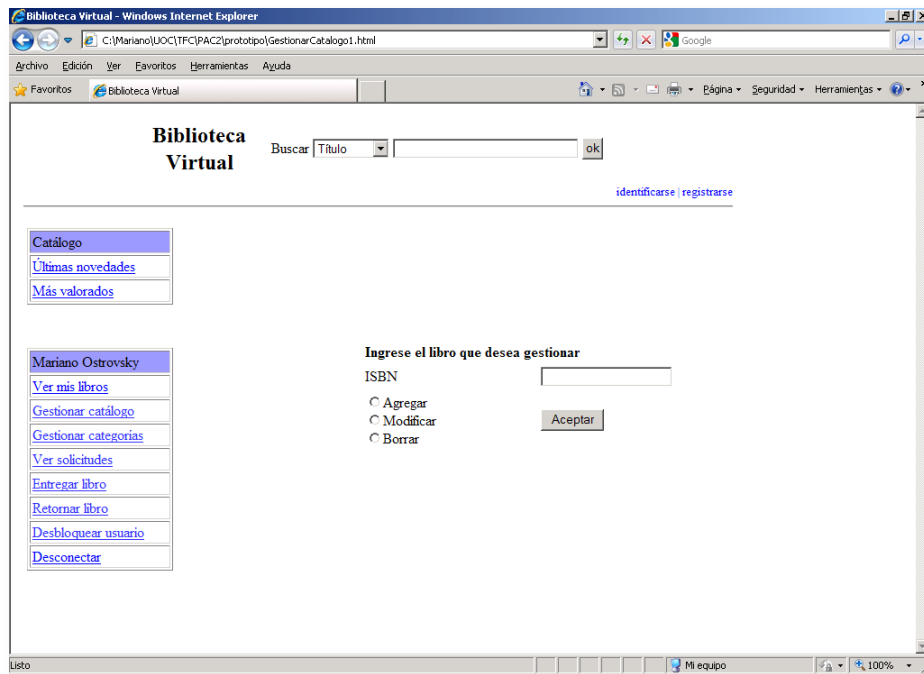


Ilustración 11. Prototipo - Gestionar catálogo

Modificar libro:

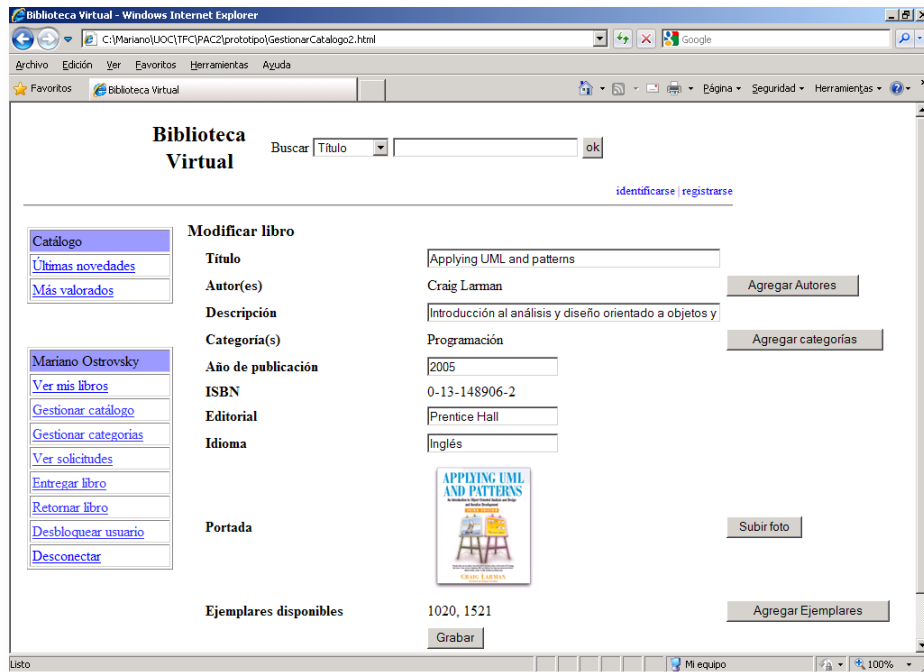


Ilustración 12. Prototipo - Modificar libro

Ver solicitudes:

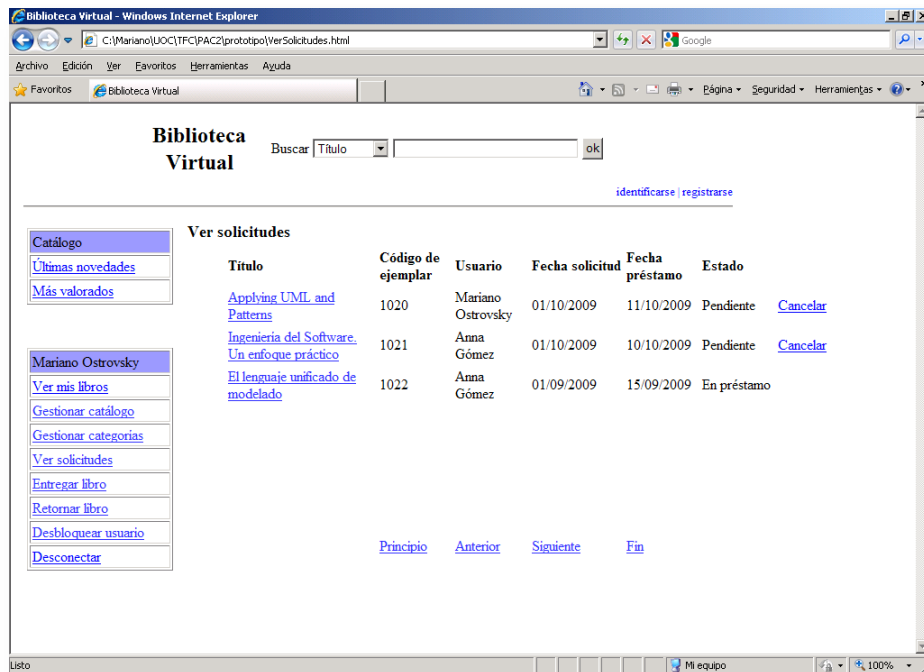


Ilustración 13. Prototipo - Ver solicitudes

Desbloquear usuario:

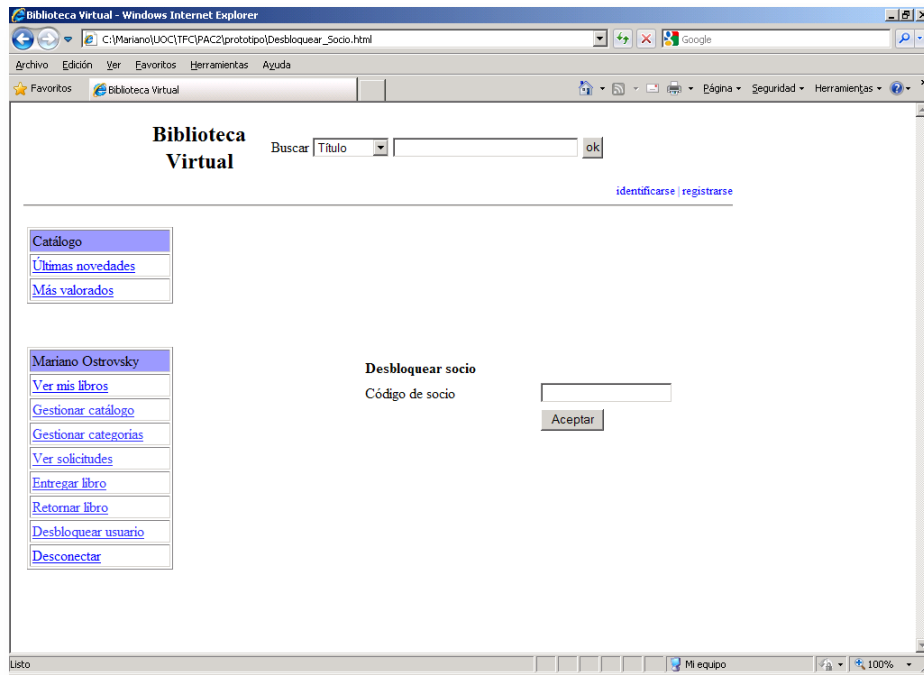


Ilustración 14. Prototipo - Desbloquear usuario

4. Diseño

Una vez acotado el problema en la fase de análisis, continuamos con el diseño de la aplicación Biblioteca Virtual presentando los diagramas de clases, estado, actividad y secuencia más significativos. También se debe realizar el diseño de la base de datos y definir la arquitectura a utilizar.

4.1. Diagrama de clases

A continuación se presenta el diagrama de clases de entidad junto con sus atributos principales:

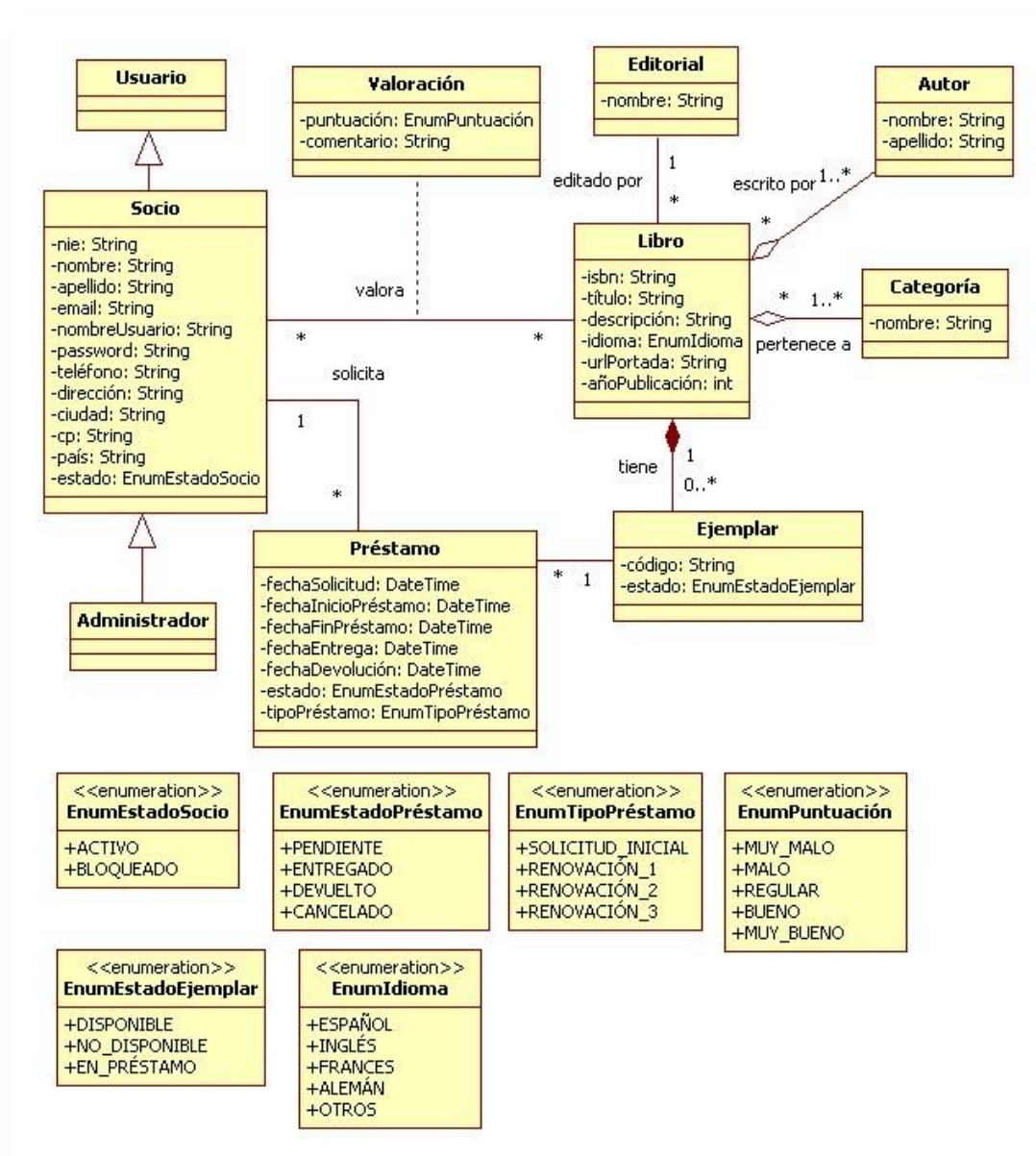


Ilustración 15. Diagrama de clases

Hay que tener en cuenta que este diagrama UML no refleja ciertas restricciones de negocio vistas en la etapa de análisis. Sin embargo, las mismas siguen siendo válidas:

- Un ejemplar no puede estar en préstamo para dos socios al mismo tiempo.
- No se puede entregar en préstamo un ejemplar que aún no ha sido devuelto.
- No se pueden solapar solicitudes de préstamos de un mismo ejemplar.
- Un préstamo no se puede renovar más de tres veces.

4.2. Diagramas de estado

Como parte del modelado de los aspectos dinámicos del sistema, se presentan los diagramas de estado más interesantes.

Estados de un socio:



Ilustración 16. Diagrama de estado - Estados de un socio

Tipos de préstamo:

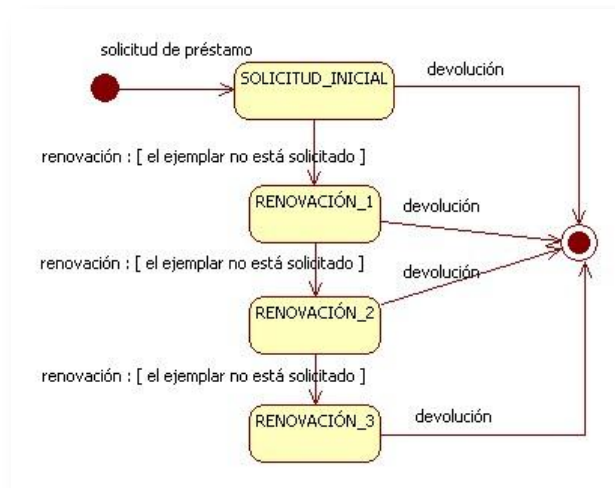


Ilustración 17. Diagrama de estado - Tipos de préstamo

Estados del préstamo:



Ilustración 18. Diagrama de estado - Estados del préstamo

Estados de un ejemplar:



Ilustración 19. Diagrama de estado - Estados de un ejemplar

4.3. Diagramas de actividad

La actividad principal de la Biblioteca Virtual es la solicitud, entrega, renovación y devolución de ejemplares. A continuación modelamos el flujo de tareas de estas actividades.

Solicitud de préstamo:

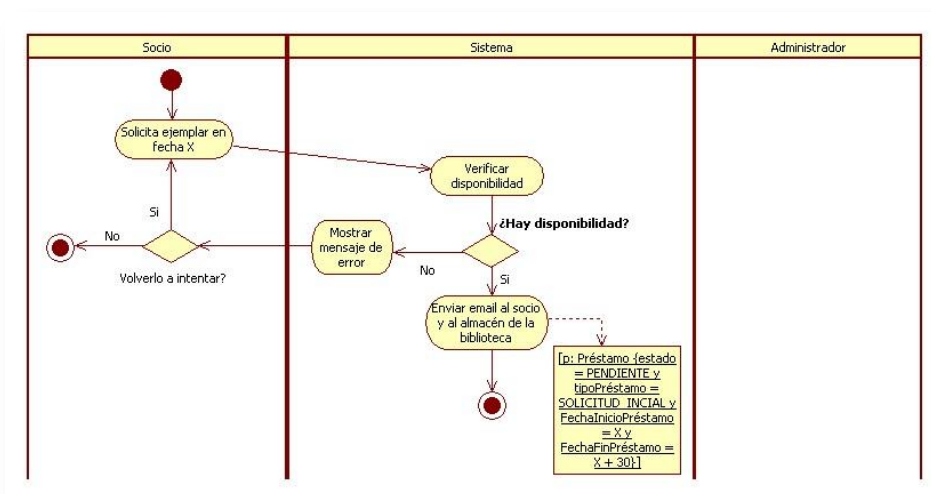


Ilustración 20. Diagrama de actividad - Solicitud de préstamo

Entrega de ejemplar:

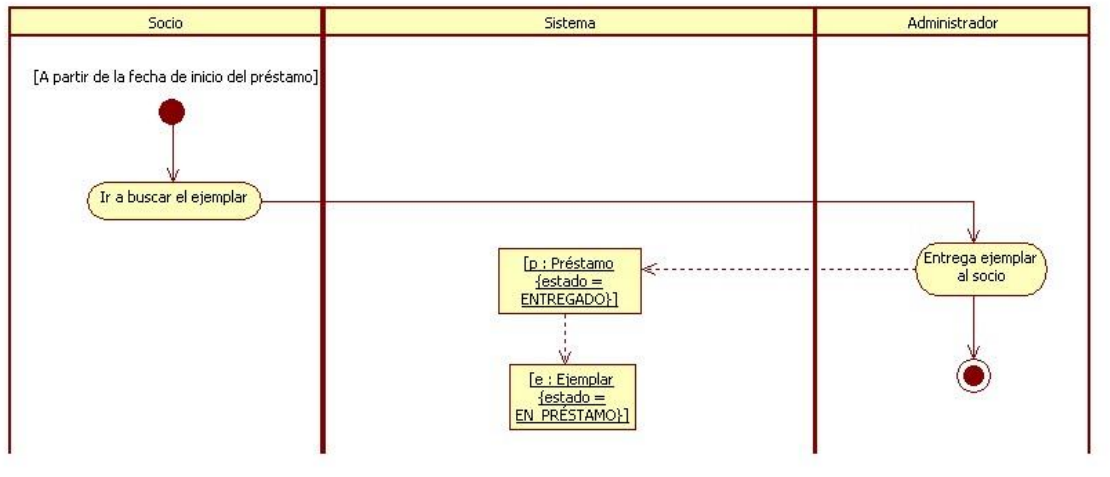


Ilustración 21. Diagrama de actividad - Entrega de ejemplar

Renovación y devolución de un ejemplar:

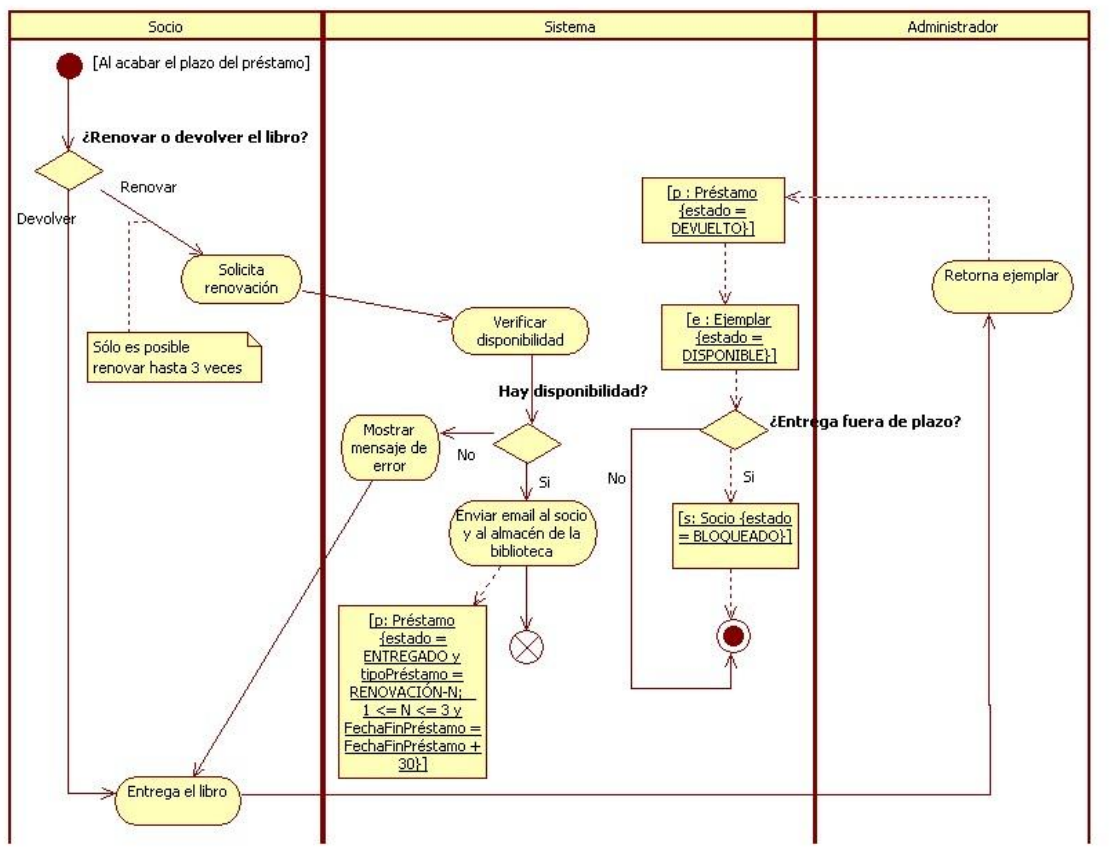


Ilustración 22. Diagrama de actividad - Renovación y devolución de un ejemplar

4.4. Diagramas de secuencia

Continuando con el modelado de los aspectos dinámicos del sistema, se presentan los diagramas de secuencia para algunos de los casos de uso más representativos de la aplicación.

Hay que tener en cuenta que es posible que en la etapa de implementación aparezcan otras colaboraciones de clase no representadas por estos diagramas. Sin embargo, estos siguen siendo válidos en tanto presentan una primera visión de las interacciones entre clases que sucederán dentro de la aplicación.

Identificación en el sistema:

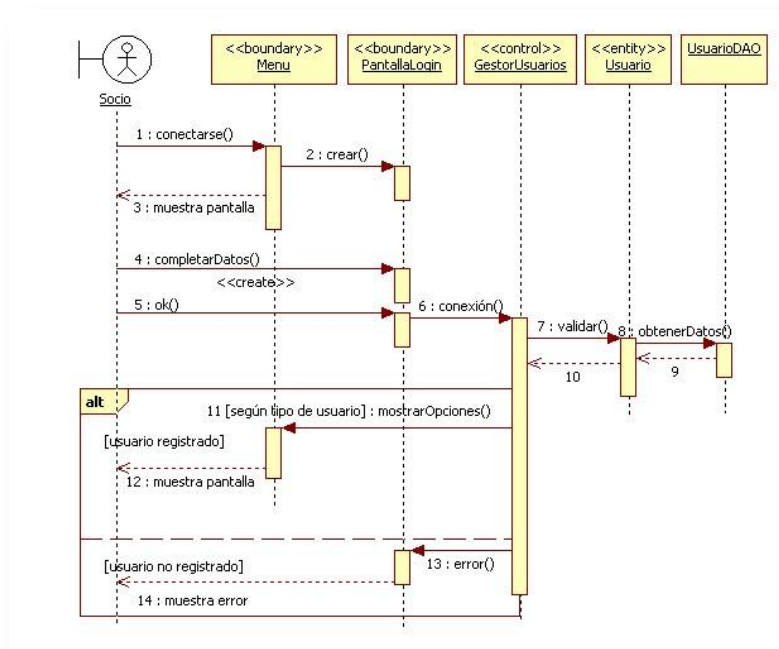


Ilustración 23. Diagrama de secuencia - Identificación en el sistema

Búsqueda de libros:

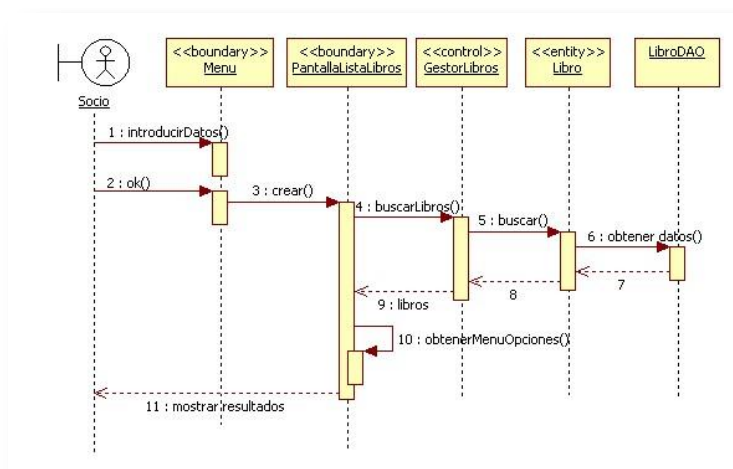


Ilustración 24. Diagrama de secuencia - Búsqueda de libros

El caso de uso *Buscar libro* es similar a los casos de uso *Ver catálogo*, *Ver mis libros* y *Ver solicitudes*. Por ello no presentamos sus diagramas de secuencia.

Registro de un nuevo socio:

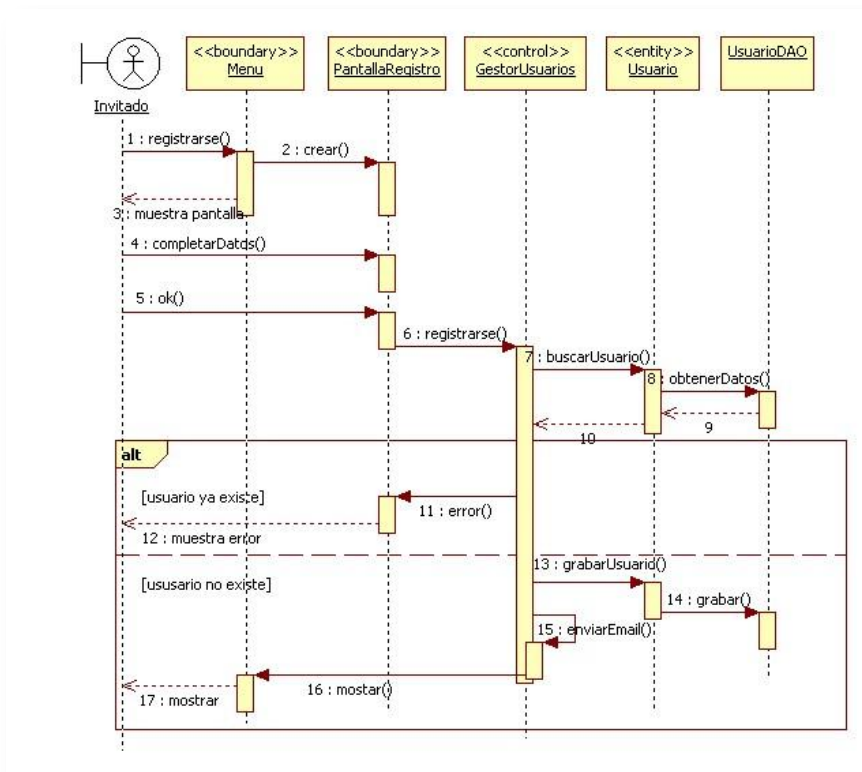


Ilustración 25. Diagrama de secuencia - Registro de un nuevo socio

Ver libro:

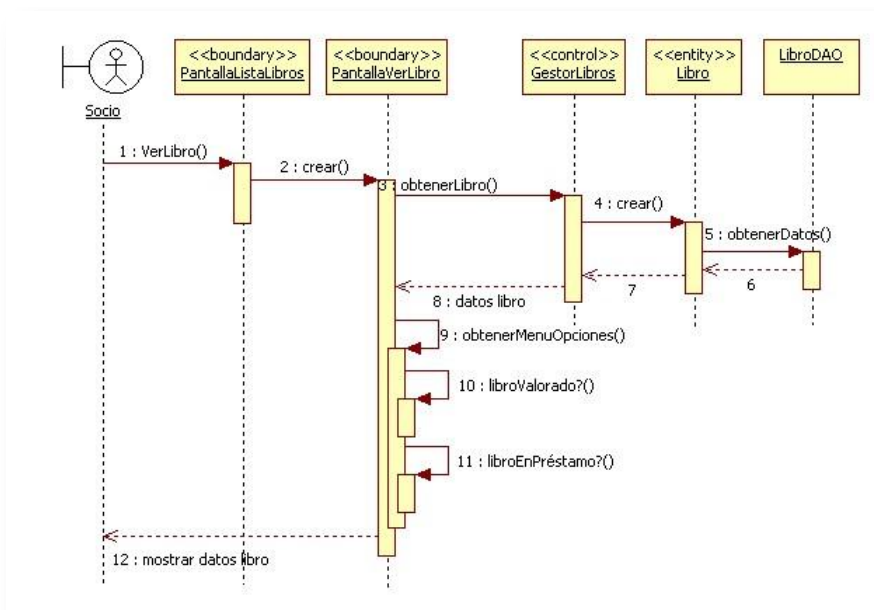


Ilustración 26. Diagrama de secuencia - Ver libro

Valorar libro:

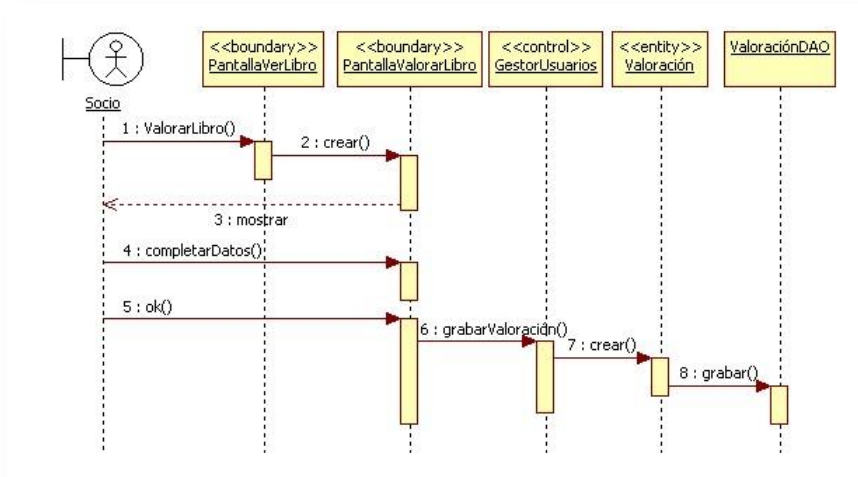


Ilustración 27. Diagrama de secuencia - Valorar libro

Solicitar préstamo:

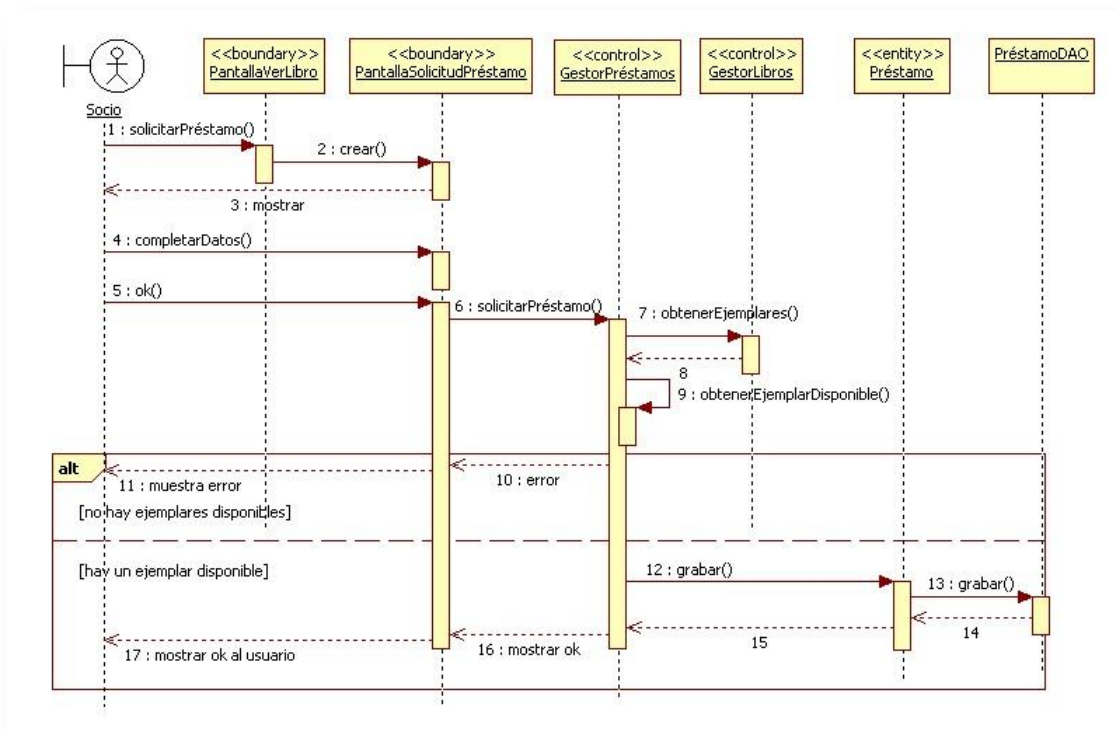


Ilustración 28. Diagrama de secuencia - Solicitar préstamo

Los casos de uso *Renovar préstamo* y *Cancelar solicitud* pueden representarse con diagramas de secuencia similares al de *Solicitar préstamo*.

Entregar libro:

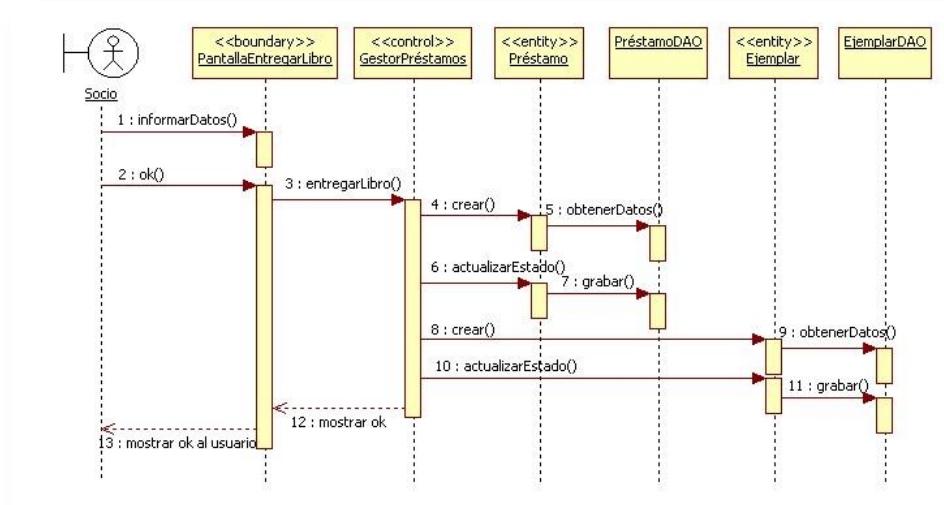


Ilustración 29. Diagrama de secuencia - Entregar libro

También aquí nos encontramos que el caso de uso *Entregar libro* es muy similar a *Retornar libro* y *Cancelar solicitud*.

4.5. Diseño de la base de datos

Utilizando notación UML para el diseño de la base de datos, obtenemos el modelo relacional normalizado:

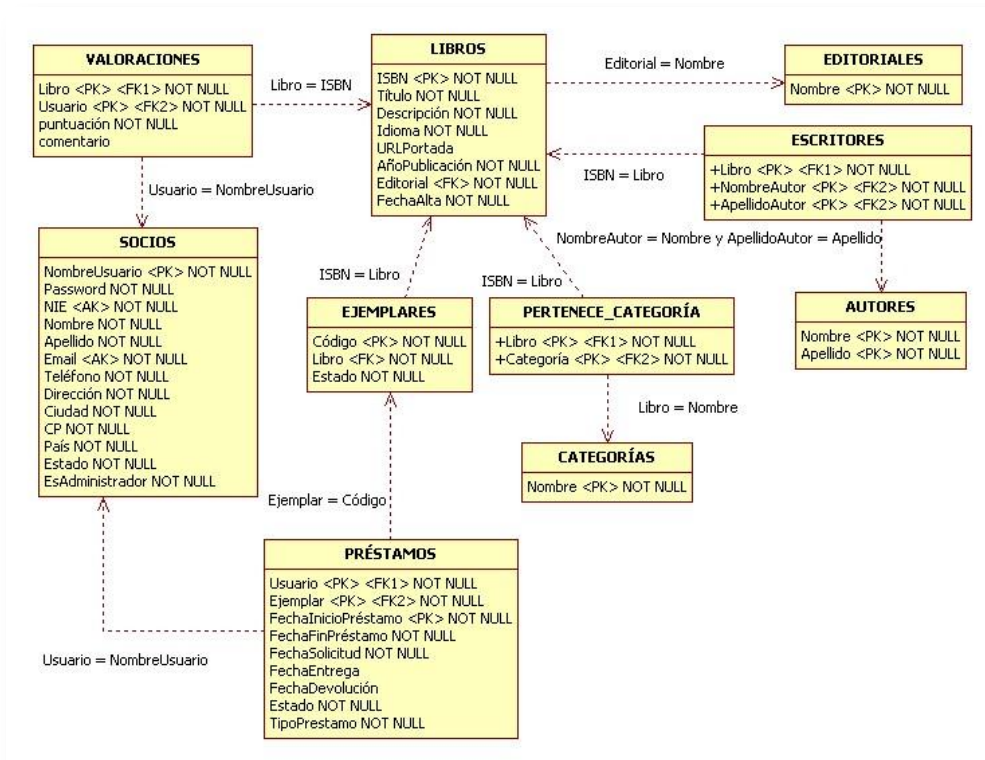


Ilustración 30. Diagrama de la base de datos

Al igual que en el diagrama estático de clases, las restricciones de negocio especificadas en la etapa de análisis que no pueden modelarse en el diagrama relacional deben tenerse en cuenta en la fase de implementación.

No se ha considerado necesario tener una entidad “Administradores” aparte, por ello se ha aprovechado la entidad “Socios” y se ha agregado un campo para especificar si el socio es también un administrador.

No se modelan los conceptos “País” o “Ciudad” como entidades separadas. Se decide no normalizar estos atributos ya que la utilidad de los mismos en el proyecto Biblioteca Virtual es muy limitada y sólo agregaría complejidad al modelo de datos.

4.6. Arquitectura de la aplicación

Teniendo en cuenta los requisitos iniciales, la arquitectura J2EE es la que mejor se adapta para el desarrollo del proyecto Biblioteca Virtual, brindando una arquitectura multicapa y distribuida, accesible a través de Internet, al sistema.

Las capas lógicas de las que estará compuesta nuestra aplicación son las siguientes:

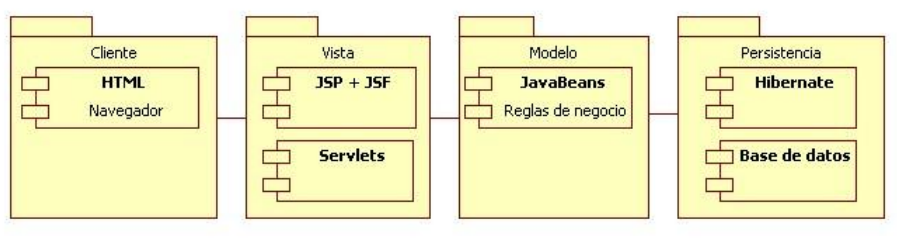


Ilustración 31. Arquitectura de la aplicación

- Cliente: se conecta a Internet y accede a la aplicación desde un navegador.
- Vista: representa la interfaz visual con la que interactuará el usuario.
- Modelo: contiene las reglas de negocio del sistema.
- Persistencia: almacena y recupera la información de un motor de base de datos.

Entre las principales ventajas de utilizar esta arquitectura multicapa encontramos:

- Mantenimiento: una modificación en una capa no implica la modificación del resto de capas de la aplicación.
- Reusabilidad: Los datos y el modelo sólo se definen una vez, permitiendo a otras aplicaciones utilizarlos sin violar las reglas del sistema.
- Escalabilidad: el sistema es flexible para dividirse físicamente cuando los requerimientos sobre el desempeño de la aplicación cambian.

- **Cliente ligero:** Los usuarios pueden conectarse a la aplicación sin importar el sistema operativo que utilizan y beneficiarse de mejoras en el sistema sin necesidad de descargar ningún software adicional.
- **Presentación desacoplada del modelo:** se pueden implementar nuevos “clientes” que accedan a la Biblioteca Virtual sin necesidad de volver a definir el modelo y las reglas de negocio.

Hay que tener en cuenta que para facilitar el desacoplamiento entre la Vista y el Modelo se utiliza el patrón de diseño Modelo-Vista-Controlador (MVC). Al utilizar MVC, los datos de negocio están separados de la lógica de presentación, y los componentes que procesan los datos no controlan la visualización de los mismos y viceversa.

Para la implementación física de esta arquitectura se han tomado las siguientes decisiones que afectan a la fase de desarrollo:

- **Java:** en el momento de escribir este TFC la última versión oficial de **J2SE** es la **6.0** y de **J2EE** la **5.0**. Se usan estas versiones para la implementación del proyecto Biblioteca Virtual. Se descarta utilizar versiones “en desarrollo” de Java.
- **Presentación:** se utilizan **JavaServer Faces (JSF 1.2)** y **JavaServer Pages (JSP 2.1)** para generar las páginas web de la aplicación e implementar el controlador de la arquitectura MVC. JSF es una tecnología reciente de J2EE, creada para simplificar el desarrollo de aplicaciones web, que provee un completo conjunto de componentes en la parte del servidor con gestión de eventos y la posibilidad de generar diferentes salidas visuales (en lenguajes basados en etiquetas como HTML o WML) para estos componentes. A diferencia de otros *frameworks* como Struts, JSF es estándar y forma parte de la especificación J2EE.
- **Modelo:** Se utilizan **JavaBeans** y **POJO** (Plain Old Java Objects) para codificar las entidades y reglas de negocio ya que nos permiten desarrollar aplicaciones ligeras en J2EE. Por otra parte, el sistema no requiere de funcionalidades más avanzadas como las que puede ofrecer el uso de Enterprise JavaBeans (EJB). Otro factor de peso para descartar el uso de EJB han sido las restricciones temporales del proyecto.
- **Persistencia:** Se utiliza **MySql** debido a que se trata de un motor de base de datos gratuito y extensamente utilizado e **Hibernate**. La ventaja de utilizar Hibernate frente a JDBC es que nos permite mapear las relaciones y atributos de nuestras tablas en objetos e independizar el código de nuestra aplicación de un cambio de la base de datos.
- **Servidor web:** Dado que no utilizamos EJB, se utiliza como servidor web **Apache Tomcat** ya que es gratuito y de los más usados. Este servidor implementa las especificaciones de servlets y JSP que necesitaremos para ofrecer el acceso a la aplicación a través de internet.

5. Implementación

Finalizada la fase de implementación, se explican las principales decisiones tomadas durante el desarrollo y los requerimientos de software del sistema. También se indican los pasos a realizar para una correcta configuración y ejecución de la aplicación Biblioteca Virtual.

Todos los casos de uso han sido implementados en esta etapa, de modo que la aplicación desarrollada es completamente funcional y cumple con todos los requisitos especificados al inicio del proyecto.

5.1. Decisiones de diseño e implementación

Durante esta fase se han realizado ajustes al diseño original producto de problemáticas no contempladas inicialmente. A su vez, se han tomado decisiones referentes a la implementación tras realizar diversas pruebas para encontrar la mejor opción que permitiese llevar a buen puerto la realización del proyecto. Entre ellas vale la pena señalar las siguientes:

- **Uso de surrogates:** Originalmente en el diseño de la base de datos no estaba previsto utilizar claves primarias sustitutas. Sin embargo, se han definido en aquellas tablas que tenían clave primaria compuesta ya que estas son complejas de utilizar desde Hibernate. Una vez finalizada la implementación hemos comprobado que uso de surrogates en Hibernate (para cualquier tipo de tabla) facilita enormemente la programación de operaciones de actualización.
- **Uso del patrón DAO:** Para abstraer y encapsular todos los accesos a la base de datos se utiliza el patrón *Data Access Object*. De esta forma, la lógica de negocio queda lo más independiente posible de la forma de almacenamiento. Aunque con Hibernate un cambio de base de datos simplemente comportaría un cambio en la configuración, este patrón de diseño nos permite qué, por ejemplo, los datos se obtengan de un web service sin grandes cambios en la aplicación.
- **Uso del patrón DTO:** Para intercambiar datos entre la capa de negocio y la capa DAO utilizamos el patrón *Data Transfer Object*. Los objetos creados por Hibernate sólo pueden utilizarse en el contexto de una sesión abierta, por ello es necesario utilizar un objeto DTO para no violar el encapsulamiento de la capa DAO. Otra posibilidad hubiese sido utilizar el patrón *Open Session In View* para tener abierta una sesión de Hibernate mientras dure la petición del usuario.
- **Framework Spring:** Este framework facilita la integración entre diferentes tecnologías Java. A pesar de sus múltiples ventajas se ha optado por no utilizarlo para, dada la ajustada planificación temporal, centrarnos en el aprendizaje de JavaServer Faces e Hibernate.
- **Interfaz de usuario:** Se ha decidió mantener la interfaz de usuario sencilla y amigable. Se ha utilizado MyFaces Tomahawk para permitir al usuario seleccionar fechas de manera visual y comprobar la integración de JavaServer Faces con otras librerías de componentes. Otras

librerías evaluadas han sido ICEfaces y RichFaces que hacen uso de Ajax y permiten desarrollar *Aplicaciones de Internet Enriquecidas*. También aquí, para poder cumplir con la planificación temporal, se ha optado por no utilizar estas tecnologías.

5.2. Diagrama de navegación

La navegación entre las páginas de la aplicación queda reflejada en el siguiente diagrama:

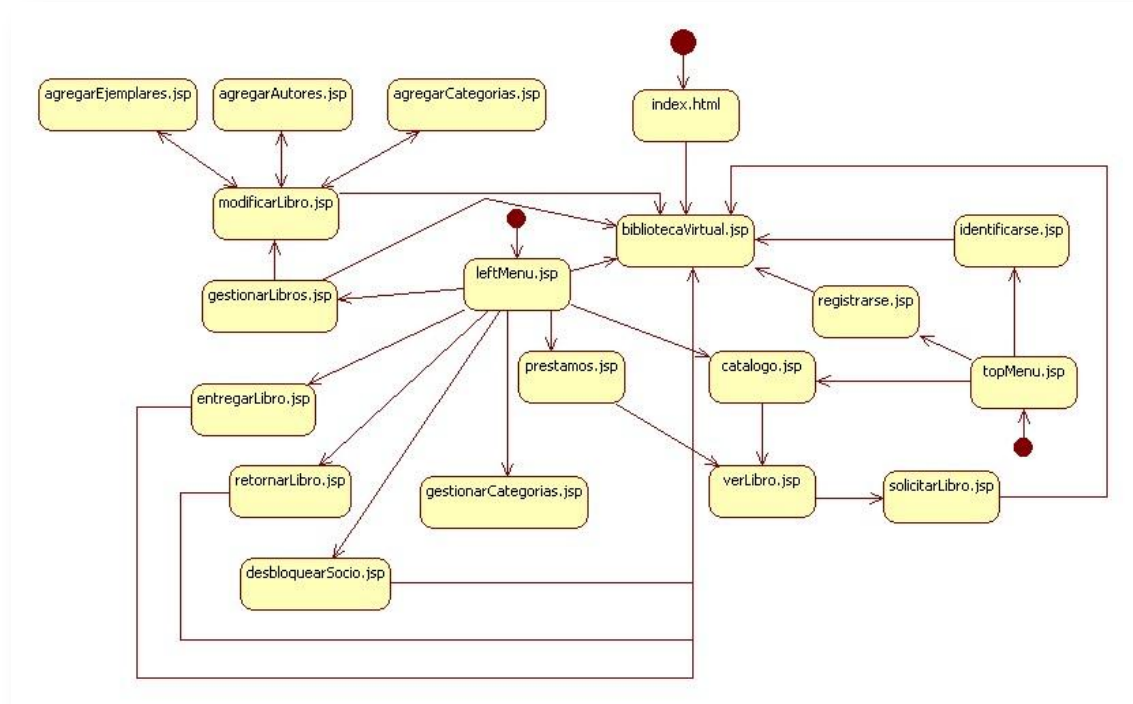


Ilustración 32. Diagrama de navegación

5.3. Diagrama de packages

Las clases de la Biblioteca Virtual se encuentran agrupadas en los siguientes packages:

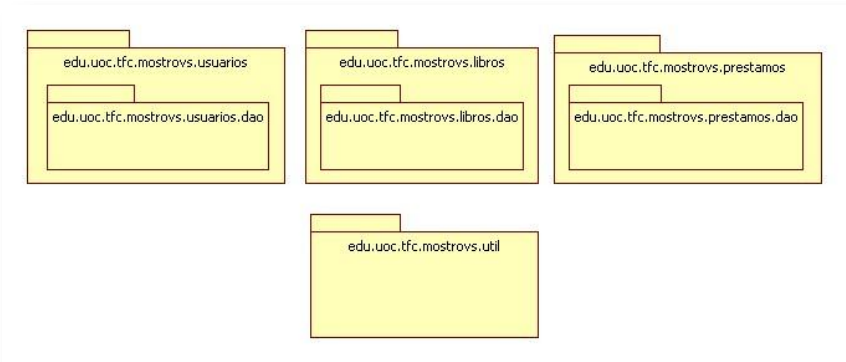


Ilustración 33. Diagrama de packages

5.4. Requerimientos de software

La aplicación necesita que las siguientes aplicaciones estén instaladas previamente en el servidor:

- **Sistema operativo Windows XP Profesional** (<http://www.microsoft.com/windows/windows-xp/default.aspx>): Si bien es posible utilizar otros sistemas operativos (versiones más modernas de Windows o Linux), la aplicación ha sido desarrollada y probada utilizando este sistema operativo.
- **Motor de base de datos MySql Community Server** (<http://www.mysql.com/>): La versión utilizada ha sido la 5.1.39. Debe realizarse la instalación estándar y configurar la contraseña del usuario root. Luego es necesario ejecutar el script de creación de la base de datos de la Biblioteca Virtual.
- **Java Enterprise Edition** (<http://java.sun.com/javaee/>): Es necesario descargar e instalar la JDK EE 5.0 y la JRE 1.6. El proceso de instalación es el estándar.
- **Servidor web Apache Tomcat** (<http://tomcat.apache.org/>): La versión utilizada ha sido la 6.0.20. El proceso de instalación es el estándar.
- **JSP Standard Tag Library Apache Standard Taglib** (<http://tomcat.apache.org/taglibs/>): Es necesario bajar la implementación 1.1.2 para poder utilizar JavaServer Pages. Deben copiarse los ficheros jstl.jar y standard.jar a la carpeta \${catalina.home}\Lib.

Adicionalmente, el sistema necesita de un conjunto de librerías que no es necesario descargar ya que las mismas se proporcionan junto al proyecto:

- **MyFaces Core 1.2.7** (<http://myfaces.apache.org/>): La implementación del proyecto MyFaces de la especificación JSF 1.2.
- **MyFaces Tomahawk 1.1.9** (<http://myfaces.apache.org/tomahawk/index.html>): Se utiliza para disponer de componentes que no están incluidos en la especificación estándar de JavaServer Faces.
- **Hibernate 3.3.2** (<http://www.hibernate.org/>): Se utiliza para facilitar el mapeo objeto-relacional entre la base de datos y la aplicación Java.
- **MySql Connector Java 5.1.10** (<http://www.mysql.com/>): Lo utiliza Hibernate para conectarse a la base de datos.
- **SLF4J 1.5.8** (<http://www.slf4j.org/>): Implementa el patrón Facade para frameworks de trazas en tiempo de ejecución. Es utilizado tanto por Hibernate como por la aplicación.

- **Apache Log4J 1.2.15** (<http://logging.apache.org/log4j/1.2/>): Es el componente elegido para dejar trazas de la ejecución del sistema.

Finalmente, cabe señalar que el usuario final sólo debe tener instalado en su ordenador un navegador web. Se ha probado la aplicación con los siguientes navegadores: Internet Explorer 8, Internet Explorer 6 y Google Chrome.

5.5. Script de creación de la Base de Datos

```
/* Si la BD existe la volvemos a crear. */
DROP DATABASE IF EXISTS biblioteca_virtual;

CREATE DATABASE biblioteca_virtual;

USE biblioteca_virtual;

CREATE TABLE Editoriales (
  nombre VARCHAR(25) NOT NULL,
  PRIMARY KEY (nombre)
) ENGINE = InnoDB;

CREATE TABLE Libros (
  isbn VARCHAR(13) NOT NULL,
  titulo VARCHAR(100) NOT NULL,
  descripcion VARCHAR(255) NOT NULL,
  idioma VARCHAR(20) NOT NULL,
  urlPortada VARCHAR(255),
  anyoPublicacion INTEGER NOT NULL,
  editorial VARCHAR(25) NOT NULL,
  fechaAlta DATETIME NOT NULL,
  PRIMARY KEY (isbn),
  FOREIGN KEY (editorial) REFERENCES Editoriales(nombre)
  ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT CHECK idioma in ('Español', 'Inglés', 'Frances',
'Alemán', 'Otros')
) ENGINE = InnoDB;

CREATE TABLE Autores (
  id INT NOT NULL auto_increment,
  nombre VARCHAR(15) NOT NULL,
  apellido VARCHAR(20) NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT UNIQUE (nombre, apellido)
) ENGINE = InnoDB;

CREATE TABLE Escritores (
  id INT NOT NULL auto_increment,
  libro VARCHAR(13) NOT NULL,
  idAutor INT NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT UNIQUE (libro, idAutor),
  FOREIGN KEY (libro) REFERENCES Libros(isbn)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (idAutor) REFERENCES Autores(id)
  ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE = InnoDB;

CREATE TABLE Categorías (
```



```
nombre VARCHAR(20) NOT NULL,  
PRIMARY KEY (nombre)  
) ENGINE = InnoDB;  
  
CREATE TABLE Pertenece_categoria (  
id INT NOT NULL auto_increment,  
libro VARCHAR(13) NOT NULL,  
categoria VARCHAR(20) NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT UNIQUE (libro, categoria),  
FOREIGN KEY (libro) REFERENCES Libros(isbn)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (categoria) REFERENCES Categorias(nombre)  
ON DELETE NO ACTION ON UPDATE CASCADE  
) ENGINE = InnoDB;  
  
CREATE TABLE Ejemplares (  
codigo VARCHAR(10) NOT NULL,  
libro VARCHAR(13) NOT NULL,  
estado VARCHAR(15) NOT NULL,  
PRIMARY KEY (codigo),  
FOREIGN KEY (libro) REFERENCES Libros(isbn)  
ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT CHECK estado in ('No disponible', 'Disponible',  
'En préstamo')  
) ENGINE = InnoDB;  
  
CREATE TABLE Socios (  
nombreUsuario VARCHAR(15) NOT NULL,  
pwd VARCHAR(10) NOT NULL,  
nie VARCHAR(10) NOT NULL,  
nombre VARCHAR(15) NOT NULL,  
apellido VARCHAR(20) NOT NULL,  
email VARCHAR(50) NOT NULL,  
telefono VARCHAR(15) NOT NULL,  
direccion VARCHAR(30) NOT NULL,  
ciudad VARCHAR(15) NOT NULL,  
cp VARCHAR(10) NOT NULL,  
pais VARCHAR(15) NOT NULL,  
estado VARCHAR(15) NOT NULL,  
esAdministrador BOOLEAN NOT NULL,  
PRIMARY KEY (nombreUsuario),  
CONSTRAINT UNIQUE (nie),  
CONSTRAINT UNIQUE (email),  
CONSTRAINT CHECK estado in ('Activo', 'Bloqueado')  
) ENGINE = InnoDB;  
  
CREATE TABLE Prestamos (  
id INT NOT NULL auto_increment,  
usuario VARCHAR(15) NOT NULL,  
ejemplar VARCHAR(10) NOT NULL,  
fechaInicioPrestamo DATE NOT NULL,  
fechaFinPrestamo DATE NOT NULL,  
fechaSolicitud DATE NOT NULL,  
fechaEntrega DATE,  
fechaDevolucion DATE,  
estado VARCHAR(15) NOT NULL,  
tipoPrestamo VARCHAR(20) NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT UNIQUE (usuario, ejemplar, fechaInicioPrestamo),  
FOREIGN KEY (usuario) REFERENCES Socios(nombreUsuario)
```

```

ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (ejemplar) REFERENCES Ejemplares(codigo)
ON DELETE NO ACTION ON UPDATE CASCADE,
CONSTRAINT CHECK estado in ('Pendiente', 'Entregado', 'Devuelto',
'Cancelado'),
CONSTRAINT CHECK tipoPrestamo in ('Solicitud inicial',
'Renovación 1', 'Renovación 2', 'Renovación 3')
) ENGINE = InnoDB;

CREATE TABLE Valoraciones (
id INT NOT NULL auto_increment,
libro VARCHAR(13) NOT NULL,
usuario VARCHAR(15) NOT NULL,
puntuacion INTEGER NOT NULL,
comentario VARCHAR(255),
fecha DATETIME NOT NULL,
PRIMARY KEY (id),
CONSTRAINT UNIQUE (libro, usuario),
FOREIGN KEY (libro) REFERENCES Libros(isbn)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (usuario) REFERENCES Socios(nombreUsuario)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT CHECK puntuacion in (1, 2, 3, 4, 5)
) ENGINE = InnoDB;

```

5.6. Instalación de la aplicación

Una vez instalados el servidor web y la base de datos estamos en condiciones de instalar la aplicación Biblioteca Virtual. La misma se encuentra empaquetada en un fichero WAR. Debemos ir al Tomcat Manager y seleccionar desplegar *TFC.war*.

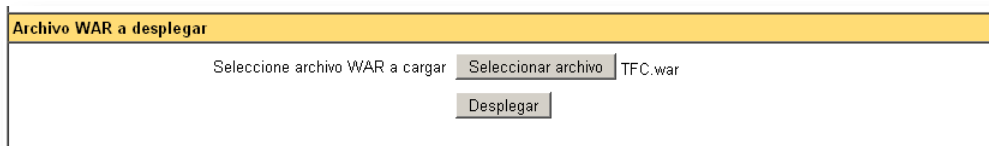


Ilustración 34. Desplegar WAR

Esta acción despliega las siguientes carpetas en el directorio $\${catalina.home}\webapps\TFC$:

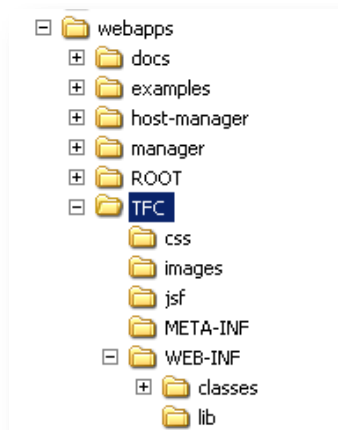


Ilustración 35. Estructura de directorios

- css: Contiene el fichero de estilos biblioteca.css
- images: Contiene la imagen por defecto cuando no hay definida una imagen de portada del libro. También contiene algunas imágenes de ejemplo.
- jsf: Contiene las páginas JSP de la aplicación.
- META-INF: contiene el fichero manifest.
- WEB-INF: contiene los ficheros de configuración web.xml y faces-config.xml
- WEB-INF/classes: contiene los packages con las clases y fuentes de la aplicación. También contiene los ficheros hibernate.cfg.xml, log4j.properties y mailServer.properties.
- WEB-INF/lib: contiene las librerías necesarias para el funcionamiento del sistema (Hibernate, Log4J, etc.)

Antes de comenzar a ejecutar la aplicación es necesario realizar algunos ajustes en la configuración.

1. En el fichero hibernate.cfg.xml hay que especificar el password del usuario root:

```
<property name="connection.url">
    jdbc:mysql://localhost/biblioteca_virtual
</property>
<property name="connection.username">root</property>
<property name="connection.password">password</property>
```

2. En el fichero mailServer.properties hay que especificar la cuenta que se utilizará para enviar los mails de la Biblioteca Virtual:

```
mail.smtp.host=smtp.proveedor.com
mail.smtp.port=25
mail.smtp.user=usuario
mail.username=usuario@proveedor.com
mail.password=password
mail.biblioteca=biblioteca@proveedor.com
mail.biblioteca.name=Biblioteca Virtual
```

3. En el fichero log4j.properties es posible cambiar donde se dejarán los logs:

```
log4j.appender.archivo.File=${catalina.base}/logs/bibliotecaVirtual.log
```

4. Finalmente, hay que dar de alta al menos un usuario administrador. Existen dos maneras de hacerlo, o bien ejecutando un INSERT o bien dando de alta un usuario normal desde la aplicación y luego actualizando el flag de administrador mediante un UPDATE:

```
INSERT INTO Socios (nombreUsuario, pwd, nie, nombre, apellido, email,
telefono, direccion, ciudad, cp, pais, estado, esAdministrador)
VALUES (:nombreUsuario, :pwd, :nie, :nombre, :apellido, :email,
:telefono, :direccion, :ciudad, :cp, :pais, 'Activo', true);

UPDATE Socios
SET esAdministrador = true
WHERE nombreUsuario = :nombreUsuario;
```

Una vez completados estos pasos el administrador puede comenzar a dar de alta categorías y libros y los socios pueden consultarlos y solicitarlos en préstamo conectándose mediante el navegador a http://ip_del_servidor:8080/TFC.

6. Conclusiones

Una de las primeras cosas que llaman la atención cuando nos acercamos al mundo Java, y en particular a J2EE, es la infinidad de especificaciones, frameworks, componentes y entornos de desarrollo que existen y conviven entre sí. A diferencia de lo que nos encontramos en otros lenguajes (como C# o Visual Basic), donde las opciones disponibles están más acotadas, al comenzar un desarrollo debemos tomar muchas decisiones iniciales que condicionarán todo el proceso de implementación. Esto se traduce en una curva de aprendizaje más larga y complicada, aunque una vez superada esta barrera inicial comprendemos que la flexibilidad, facilidad de uso y potencia hayan hecho que J2EE sea una de las principales opciones de desarrollo en el mundo empresarial.

Otra cosa que llama la atención es la estrecha relación que hay entre los patrones de diseño y el mundo Java. Probablemente el hecho de ser una tecnología moderna ha facilitado que se hayan tenido en cuenta los conocimientos adquiridos a lo largo del tiempo sobre buenas prácticas de diseño y programación. Por ejemplo, el soporte al patrón MVC está presente en prácticamente todos los frameworks disponibles para aplicaciones web y la mayoría de componentes utilizan los patrones *Abstract Factory*, *Singleton*, *Facade*, etc.

Respecto a las tecnologías utilizadas, cabe decir que JavaServer Faces se ha mostrado como una excelente opción a la hora de desarrollar aplicaciones web. Ha permitido separar de forma clara la capa de presentación de la lógica de negocio y crear de manera sencilla nuevos componentes, validadores, etc. Además, es una buena opción para trabajar con un equipo de diseñadores y programadores.

En el caso de Hibernate, su uso ha obligado a reconsiderar la forma de obtener los datos y el modo en que éstos se propagan por la aplicación. La configuración es compleja pero, una vez realizada, facilita el mapeo objeto relacional y su lenguaje de consultas (HSQL) es muy similar al SQL.

También hay que mencionar que sin una planificación clara y realista hubiese sido imposible llevar el trabajo a buen puerto. A su vez, hemos comprobado que utilizar una metodología rigurosa de análisis y diseño orientado a objetos permite abstraernos en gran parte del lenguaje utilizado y facilita enormemente la posterior implementación.

Finalmente, podemos afirmar que el desarrollo del proyecto Biblioteca Virtual ha resultado una experiencia enriquecedora y que todos los objetivos propuestos al comienzo del TFC han sido conseguidos.

Glosario

A continuación presentamos los términos utilizados en la comprensión del análisis del proyecto.

Administrador: Usuario del sistema con privilegios especiales encargado de gestionar el catálogo, entregar y recibir ejemplares en préstamo.

Biblioteca Virtual: Se trata de la aplicación que implementa el proyecto Biblioteca Virtual.

Catálogo: Son los libros disponibles en la biblioteca para préstamo de los socios.

Ejemplar: Es un libro que existe en el mundo real, está dado de alta en el catálogo y puede ser prestado. Cada ejemplar tiene un código que lo hace único.

Invitado: Usuario del sistema con permisos para consultar el catálogo sin necesidad de identificarse.

ISBN: El *International Standard Book Number* es un identificador único de libros adoptado como estándar internacional.

Libro: En el contexto del sistema puede identificar de manera genérica una obra impresa o bien un ejemplar existente en la biblioteca.

Sistema: La aplicación Biblioteca Virtual.

Socio: Usuario del sistema con permisos para solicitar libros en préstamo.

Solicitud: Petición de préstamo de un libro.

Stock: Disponibilidad real de un libro del catálogo.

Préstamo: Libro en poder del usuario que pertenece a la biblioteca.

Usuario: Persona que se conecta al sistema. Puede hacerlo con el rol de Invitado, Socio o Administrador.

Bibliografía

Booch, Grady; Rumbaugh, James; Jacobson, Ivar (1999). El lenguaje unificado de modelado. Addison Wesley.

Bauer, Christian; King, Gavin (2005). Hibernate In Action. Manning.

Cockburn, Alistair (2001). Writing Effective Use Cases. Addison-Wesley.

Connolly, Thomas; Begg, Carolyn (2005). Database Systems: A Practical Approach to Design, Implementation, and Management (4ª edición). Addison Wesley.

Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John (1995). Patrones de Diseño. Addison Wesley.

Geary, David (2002, 29 de Noviembre). “A first look at JavaServer Faces, Part 1”. Javaworld.
<http://www.javaworld.com/javaworld/jw-11-2002/jw-1129-jsf.html>.

Geary, David (2002, 27 de Diciembre). “A first look at JavaServer Faces, Part 2”. Javaworld.
<http://www.javaworld.com/javaworld/jw-12-2002/jw-1227-jsf2.html>.

Larman, Craig (2005). Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3ª edición). Prentice Hall.

Mahmoud, Qusay H. (2004, Agosto). “Developing Web Applications with JavaServer Faces”. Sun Developer Network.
<http://java.sun.com/developer/technicalArticles/GUI/JavaServerFaces/>.

Mann, Kito D. (2004, 13 de Diciembre). “Getting around JSF: The role of JSP”. Javaworld.
<http://www.javaworld.com/javaworld/jw-12-2004/jw-1213-jsf.html>.

Mukhar, Kevin; Zelenak, Chris (2006). Beginning Java EE 5 Platform: From Novice to Professional. Apress.

Murach, Joel; Steelman, Andrea (2008). Java Servlets and JSP (2ª edición). Murach.

Pérez García, Alejandro (2005, 15 de Agosto). “JSF - JavaServer Faces (y comparación con Struts)”. AdictosAlTrabajo.com.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jsf>.

Prakash Joshi, Anand (2005, 2 de Diciembre). “Design with the JSF architecture”. IBM DeveloperWorks.
<http://www.ibm.com/developerworks/java/library/wa-dsgnpatjsf.html?ca=drs->

Pressman, Roger S. (2006). Ingeniería del software: Un enfoque práctico (6ª edición). McGraw-Hill.

Roman, Ed; Patel Sriganesh, Rima; Brose, Gerald (2005). Mastering Enterprise JavaBeans (3ª edición). Wiley.

Sun Microsystems (2001-2002). “Core J2EE Patterns”. Sun Developer Network.
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>.

Sun Microsystems (2007). The Java EE 5 Tutorial: For Sun Java System Application Server 9.1.
Sun Microsystems.

Zambon, Giulio; Sekler, Michael (2007). Beginning JSP, JSF, and Tomcat Web Development:
From Novice to Professional. Apress.

Anexos

Imágenes del sitio web Biblioteca Virtual

A continuación se muestran algunas capturas de pantalla del sitio web desarrollado. Podemos observar que se ha respetado la idea original plasmada en el prototipo con algunos retoques producto de la implementación realizada.

Registro de un nuevo socio:

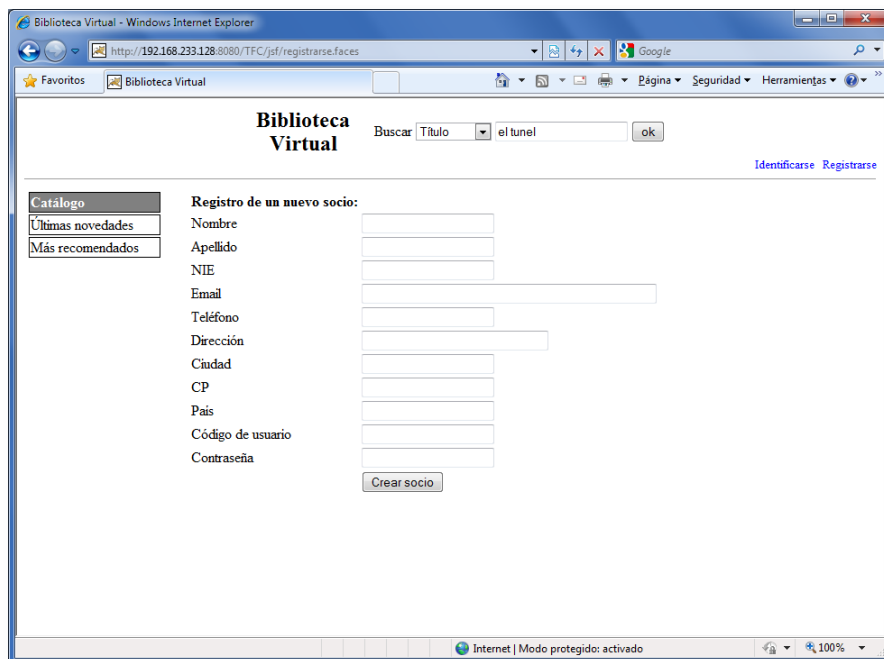


Ilustración 36. Biblioteca Virtual - Registro de un nuevo socio

Conexión al sistema:

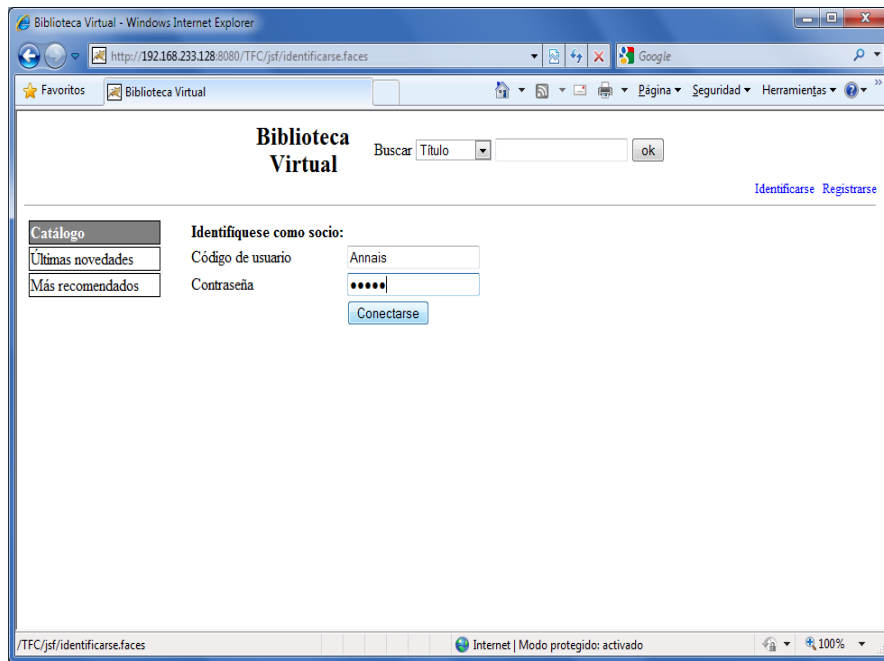


Ilustración 37. Biblioteca Virtual - Conexión al sistema

Búsqueda de últimas novedades:

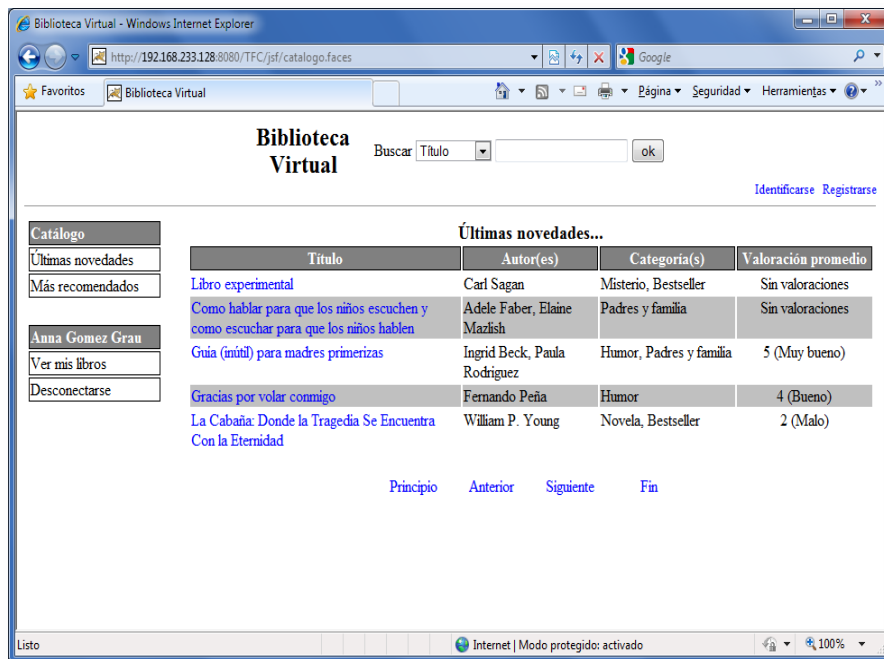


Ilustración 38. Biblioteca Virtual - Búsqueda de últimas novedades

Ver libro:

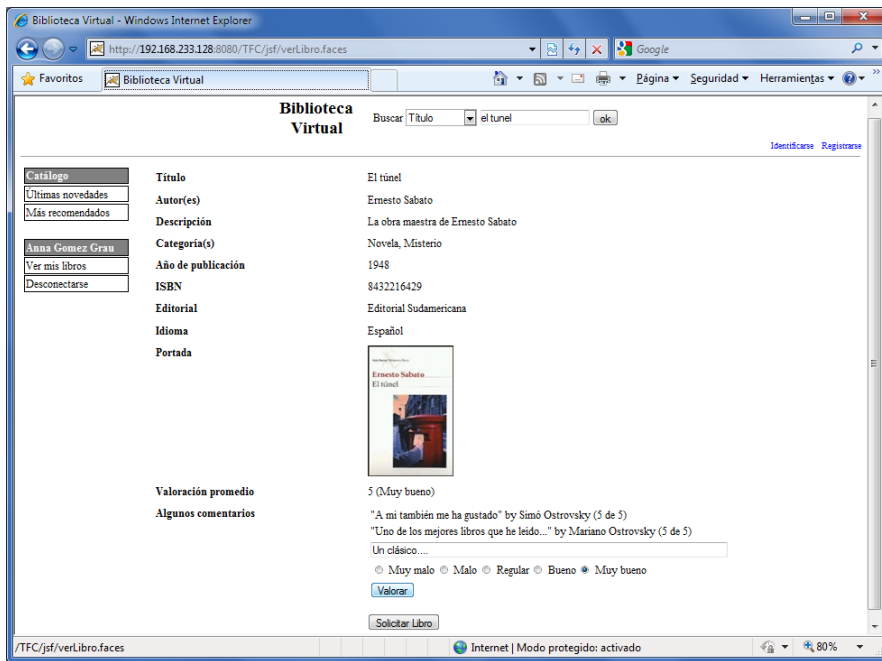


Ilustración 39. Biblioteca Virtual - Ver libro

Elección de fecha al solicitar el libro:

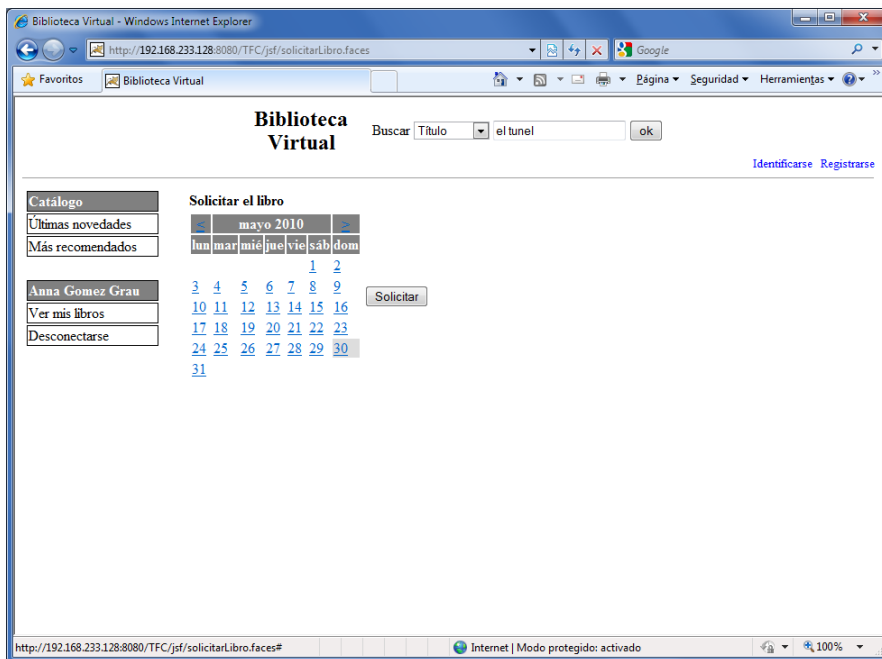


Ilustración 40. Biblioteca Virtual - Elección de fecha al solicitar el libro

Ver mis libros:

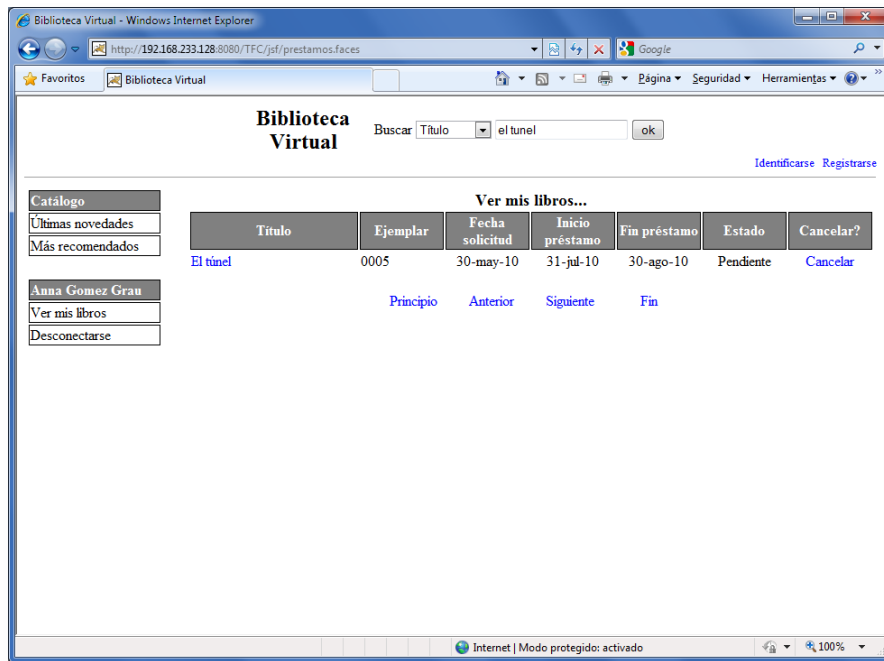


Ilustración 41. Biblioteca Virtual - Ver mis libros

Ver solicitudes (conectado como administrador):

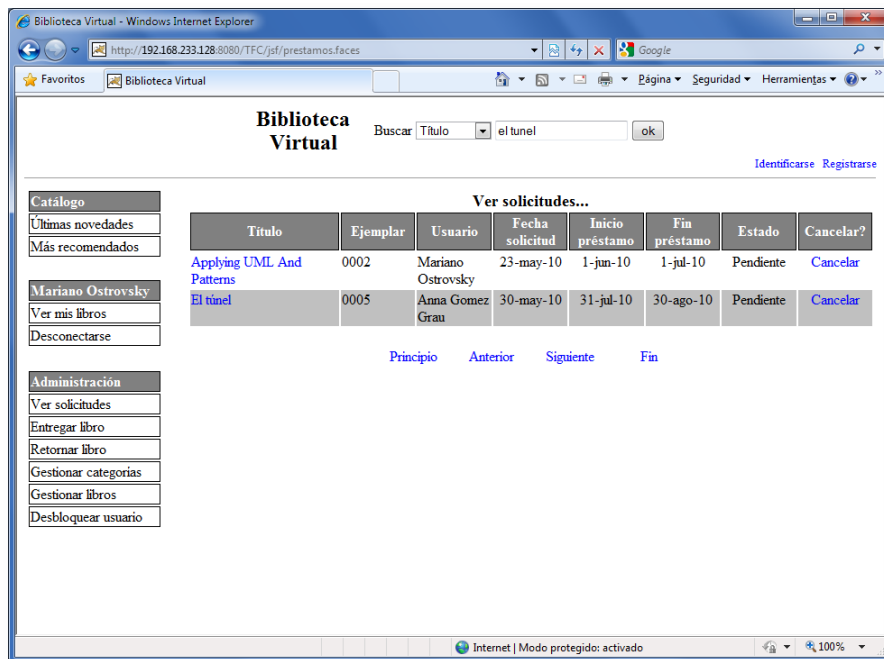


Ilustración 42. Biblioteca Virtual - Ver solicitudes

Entregar libro:

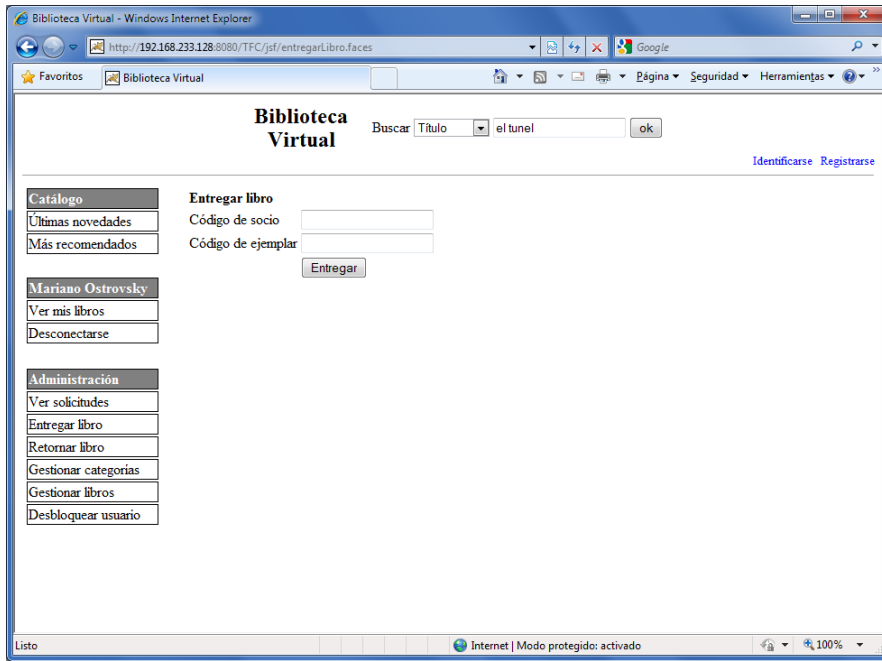


Ilustración 43. Biblioteca Virtual - Entregar libro

Retornar libro:

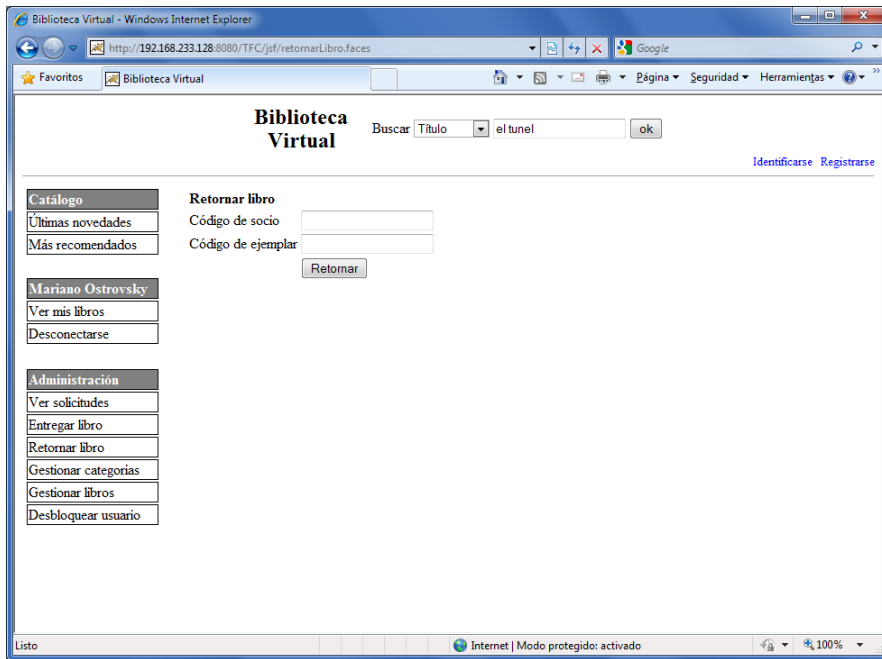


Ilustración 44. Biblioteca Virtual - Retornar libro

Gestionar categorías:

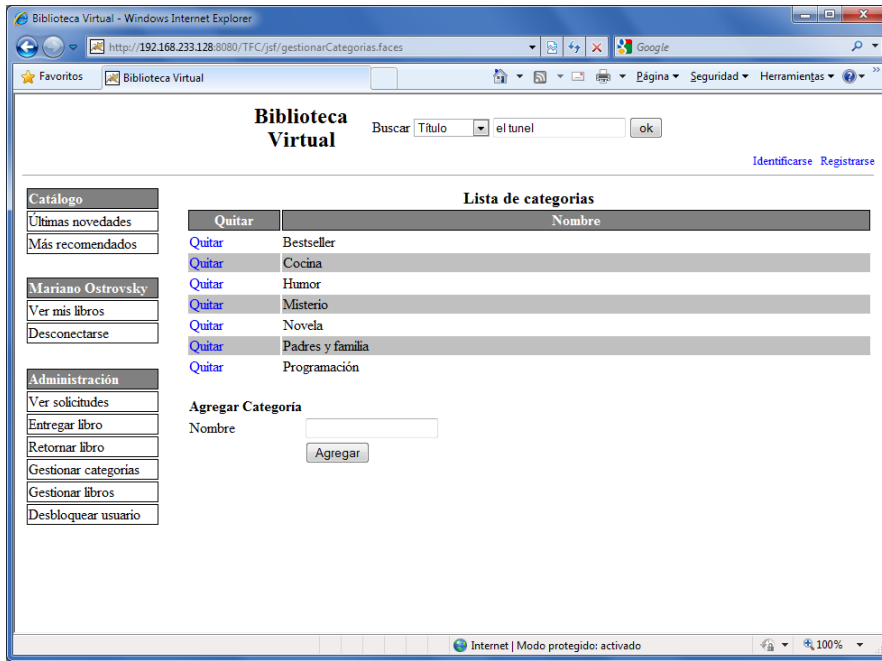


Ilustración 45. Biblioteca Virtual - Gestionar categorías

Gestionar libros:

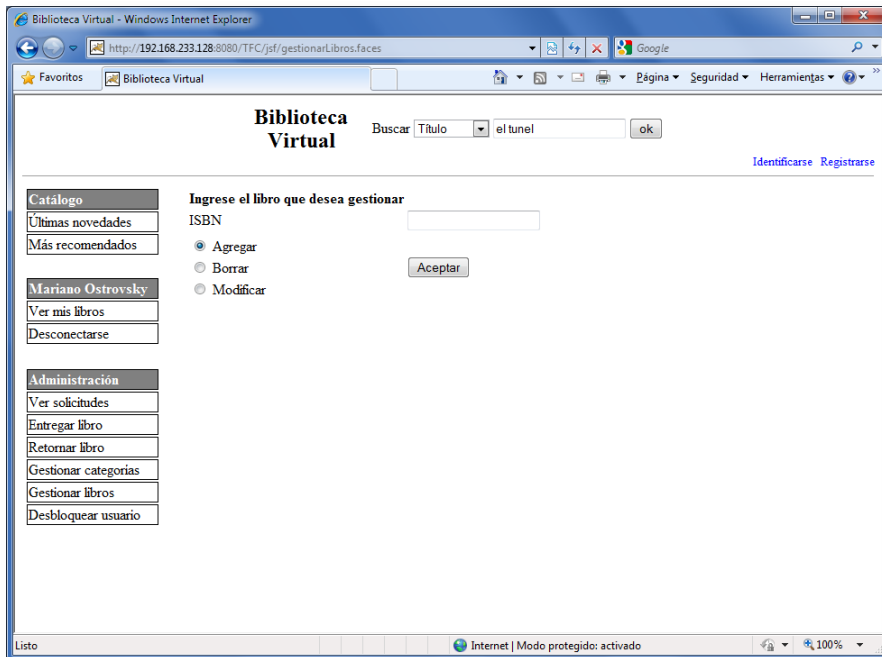


Ilustración 46. Biblioteca Virtual - Gestionar libros

Modificar datos del libro:

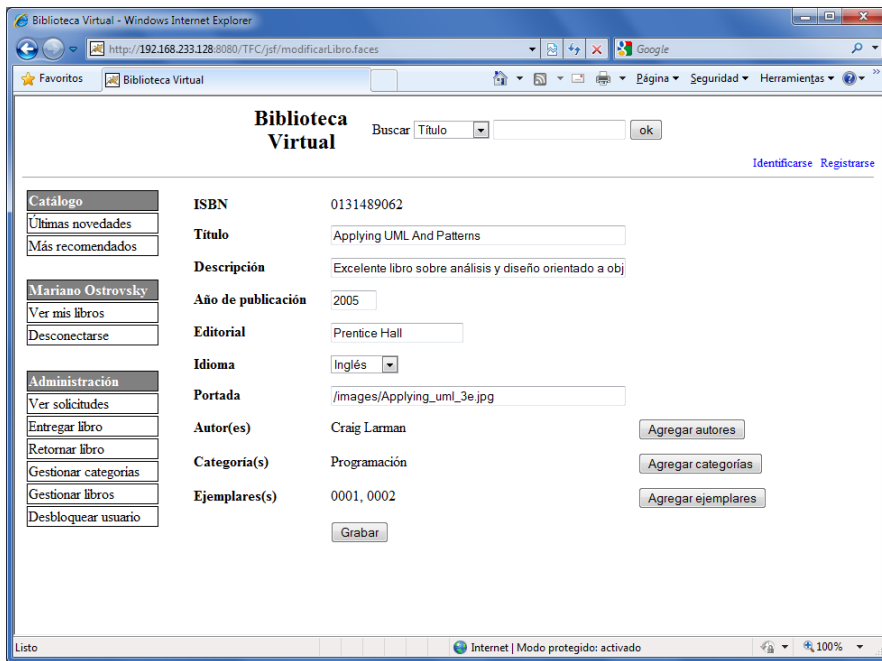


Ilustración 47. Biblioteca Virtual - Modificar datos del libro

Modificar autores del libro:

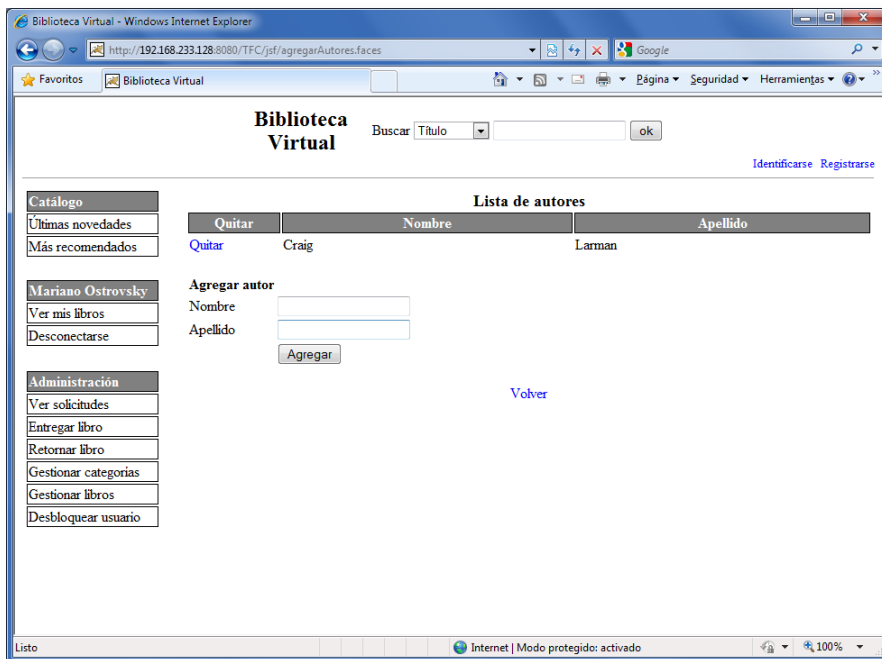


Ilustración 48. Biblioteca Virtual - Modificar autores del libro

Modificar categorías del libro:

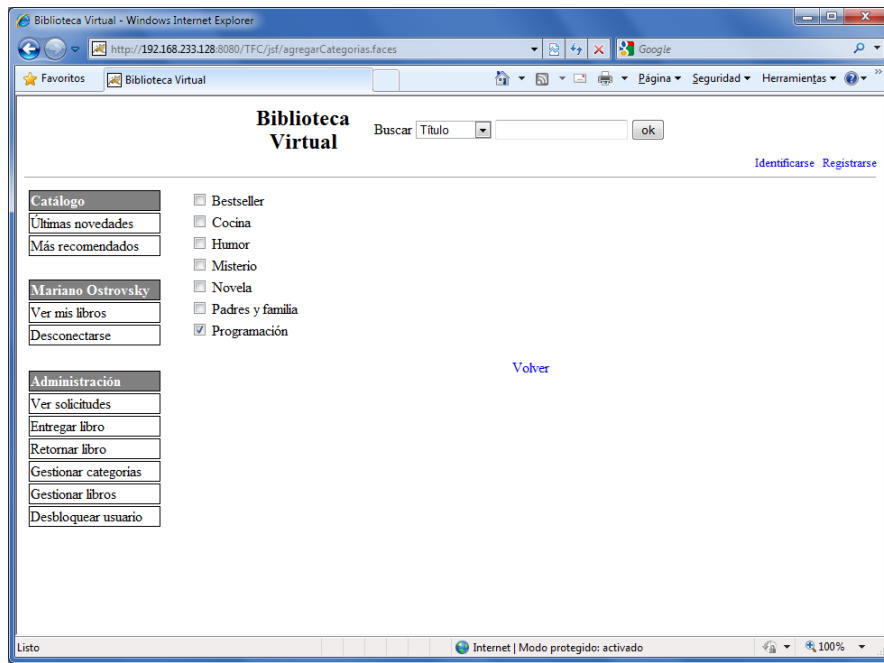


Ilustración 49. Biblioteca Virtual - Modificar categorías del libro

Modificar ejemplares del libro:

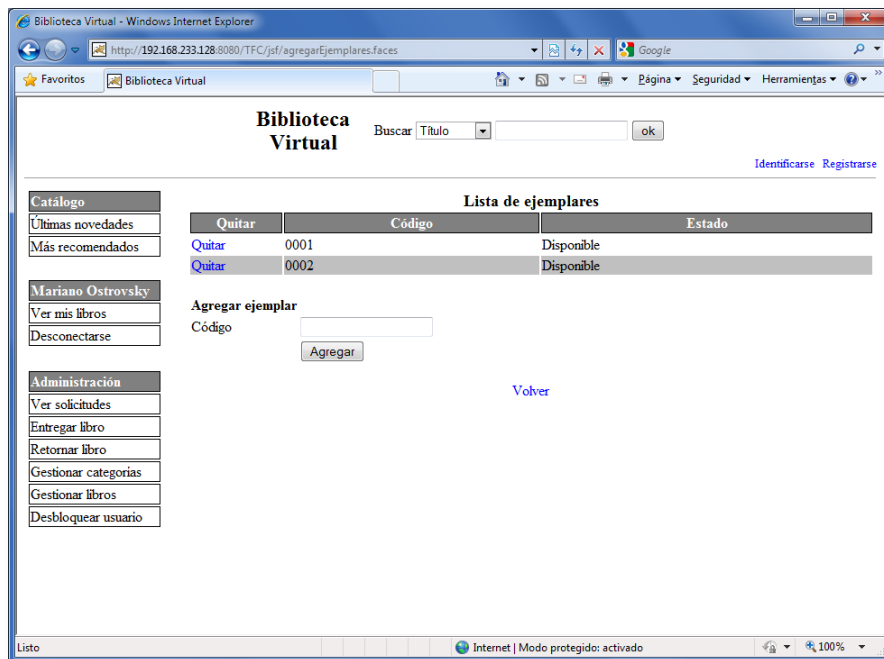


Ilustración 50. Biblioteca Virtual - Modificar ejemplares del libro

Desbloquear socio:

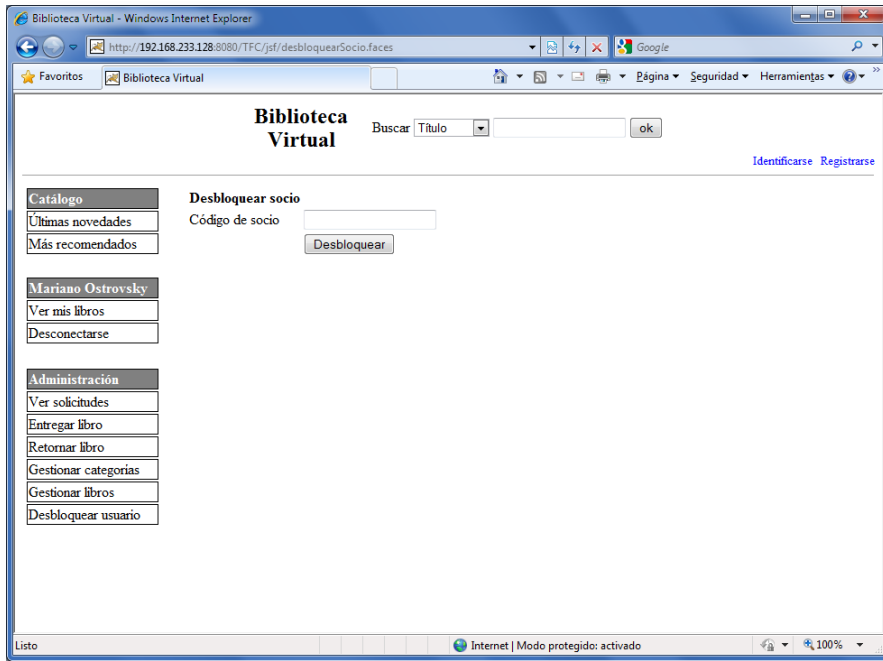


Ilustración 51. Biblioteca Virtual - Desbloquear socio