



APLICACIÓN BASE PARA EL DESARROLLO EN J2EE

Guia paso a paso de documentación de especificaciones de desarrollo

Edición: V 1.0

**Autor: Unidad de Informática Corporativa
Área de Desarrollo y Mantenimiento**

Fecha: Diciembre de 2004

Historia del Documento

Versión: V 1.0	<i>Descripción:</i>		
	<i>Elaborado por:</i>	Juan José Barrio Vizán	<i>Fecha:</i> Diciembre 2004
	<i>Revisado por:</i>	Diego García Carrera	<i>Fecha:</i> Enero 2005
			<i>Fecha:</i>
			<i>Fecha:</i>
	<i>Aprobado por:</i>	Juan A. Barras García	<i>Fecha:</i> Enero 2005

Lista de distribución del documento

Nombre	Localización

Índice

1	Introducción	4
2	Catálogo de requisitos	6
3	Modelo del Negocio (diagrama de actividades)	7
4	Modelo del dominio	8
5	Entorno tecnológico	9
6	Modelo de casos de uso.....	10
7	Análisis de los casos de uso, diagrama de clases de análisis y detalle de pantallas	12
7.1	Análisis de los casos de uso	12
7.2	Diagrama de clases de análisis.....	14
7.3	Detalle de pantallas	16
8	Realizaciones de diseño.....	17
9	Clases de diseño.....	19
10	Base de datos.....	21
11	Cuadernos de carga.....	22
11.1	Descripción.....	22
11.2	Diferencias en el nivel de detalle en las plantillas	23

1 Introducción

Este documento se va a dedicar al estudio de los diferentes aspectos de análisis y diseño que se deben completar para llegar a la construcción de los cuadernos de carga que se pasarán a los programadores para el desarrollo del software J2EE en la Junta de Castilla y León.

Va dirigido a aquellas personas que se encargan del análisis y diseño de aplicaciones J2EE en JCyL. Deben tener conocimientos en UML y aplicaciones de tres capas.

Para ilustrar los procesos de forma sencilla, vamos a suponer una aplicación que se debe implementar y vamos a ver los distintos productos que vamos obteniendo hasta llegar a los cuadernos de carga. Para todo el proceso de desarrollo del software vamos a seguir los pasos indicados por Métrica 3.

Consideremos el siguiente ejemplo:

Queremos construir un sistema de gestión de libros (GELI). En este sistema se van a considerar diferentes entidades. Como es lógico, tendremos los libros, que contendrán información acerca del título, breve descripción, autor, género y fecha de edición. Por otro lado estarán los autores de los libros, para los que se va a almacenar información acerca de su nombre, apellidos, lugar de nacimiento. Finalmente vamos a disponer de una serie de géneros literarios según los que se van a clasificar los libros.

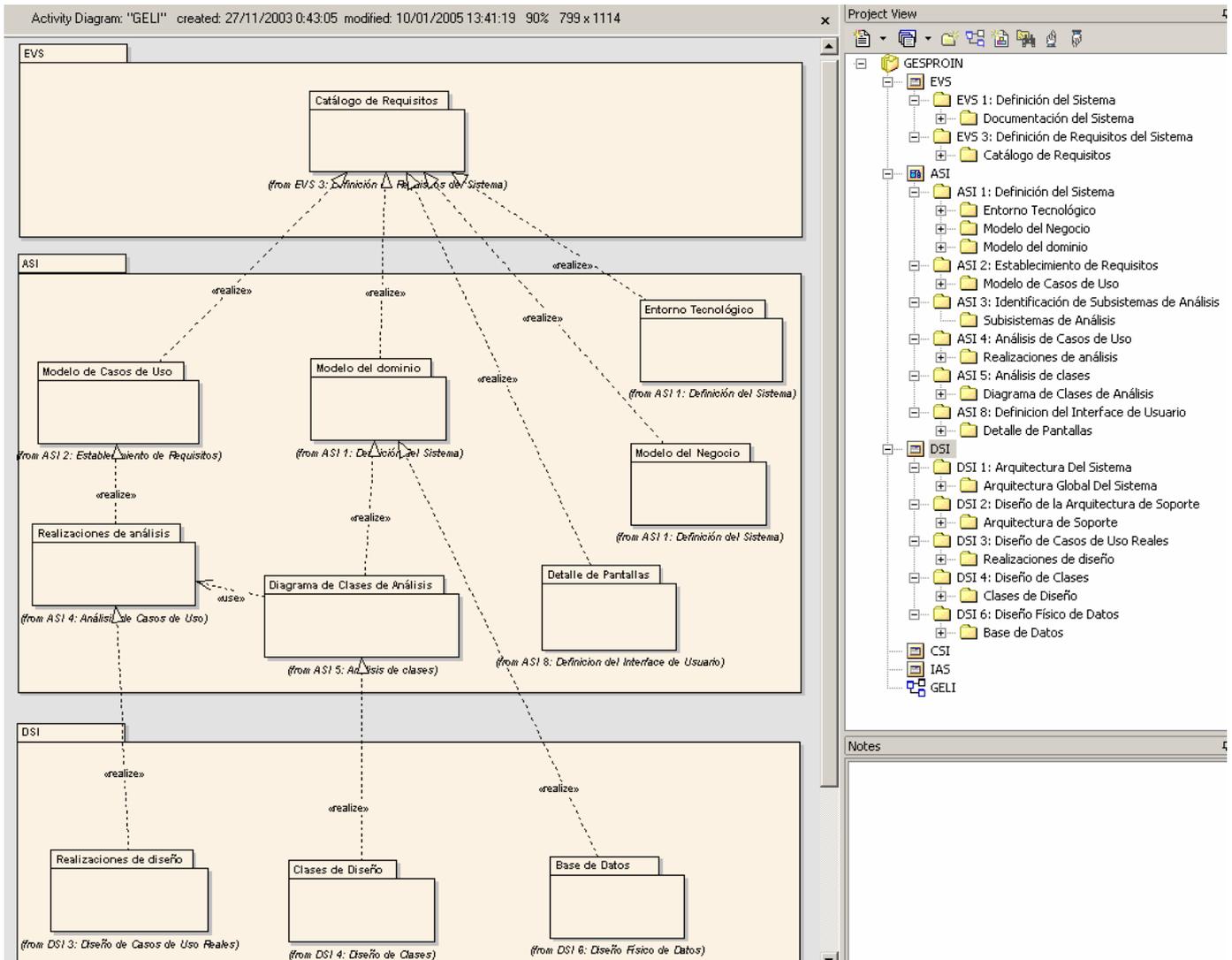
Esta aplicación va a ser utilizada por dos tipos de personas

- Administradores: los usuarios de este grupo podrán realizar el mantenimiento de todas estas entidades. Además podrán realizar consultas acerca de los libros del sistema (en este caso se va a permitir buscar libros por cualquiera de sus campos), los autores, y para cada autor podrán ver todos los libros disponibles. Estos listados podrán obtenerse en formato PDF y Excel.
- Operadores: los usuarios pertenecientes a este grupo sólo podrán realizar consultas de información que se haya dado de alta en el sistema. Igual que en el caso anterior podrán obtener estos listados en formato PDF y Excel.

La aplicación se ejecutará en un entorno Web y necesitará la validación de los usuarios para comprobar si el usuario es de algún tipo de los que hemos identificado.

Este es, a grandes rasgos, el sistema que se va a construir a lo largo de este documento.

Como herramienta para documentar todo el proceso vamos a utilizar Enterprise Architect v4.0 (EA). Para esta herramienta se dispone de una plantilla general en la que se ven los diferentes apartados que se pueden desarrollar, siguiendo las indicaciones de Métrica 3.



En la imagen se muestran todos los productos que se pueden obtener en el proceso de desarrollo de software, pero que algunos de ellos no se consideren necesarios va a depender, sobre todo, de la complejidad del sistema con el que se vaya a trabajar.

La aplicación sobre la que vamos a desarrollar este documento es sencilla y por tanto, habrá apartados de esta plantilla que no generaremos. Junto con este documento se entregará el fichero, en formato EA, con toda la documentación generada para nuestra aplicación.

Cada uno de los apartados vamos ver que se pueden detallar más o menos en función de la persona o personas a quien vaya dirigido el documento.

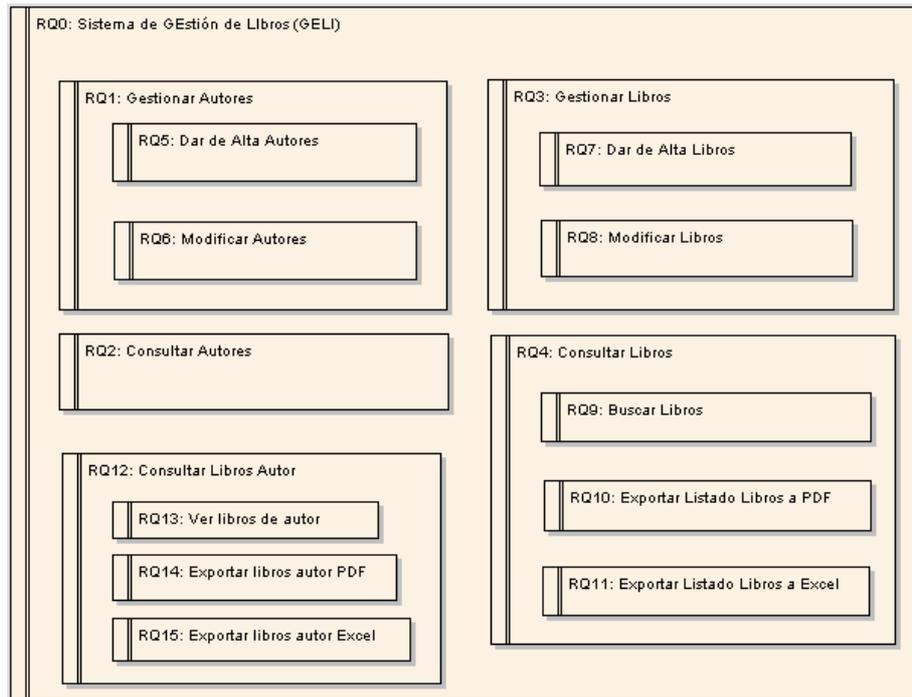
Consideraremos tres niveles de usuarios

- **Nivel 1:** usuarios internos a la organización con poca experiencia en desarrollo de aplicaciones J2EE
- **Nivel 2:** Usuarios internos a la organización con experiencia en desarrollo de aplicaciones J2EE
- **Nivel 3:** Usuarios externos a la organización

2 Catálogo de requisitos

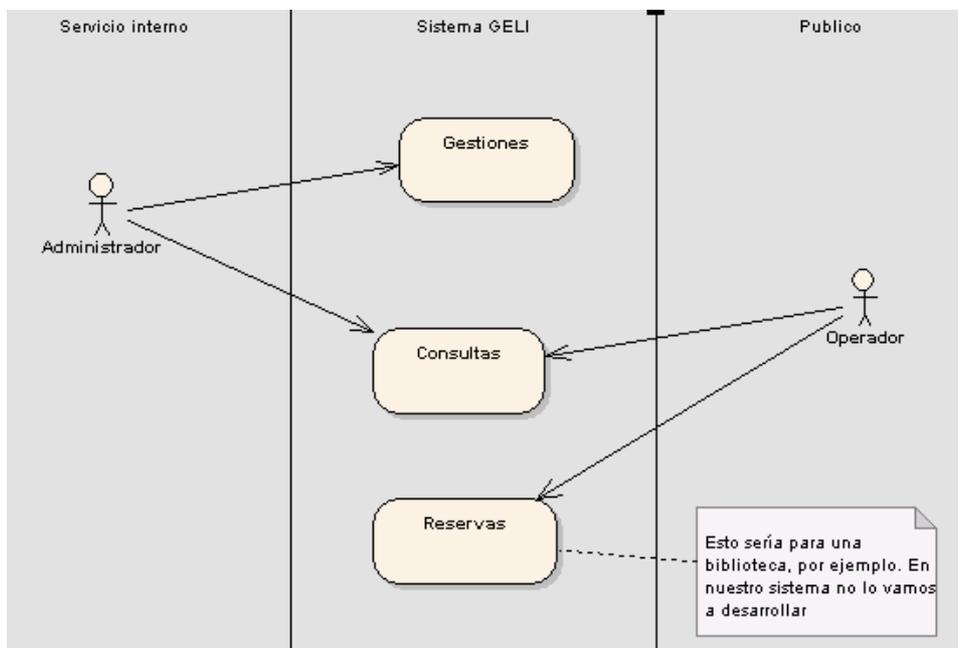
Para comenzar el análisis del sistema, una vez obtenidas las especificaciones, habrá que identificar los requisitos. Para ello vamos a empezar a construir un catálogo con todos los requisitos identificados. Este catálogo podrá ser modificado a medida que se vayan completando las diferentes fases del análisis y diseño.

Utilizando EA para definirlos, se llega a una situación parecida a la siguiente



3 Modelo del Negocio (diagrama de actividades)

Una vez vistos los requisitos, vamos a identificar la secuencia de actividades que se van a producir para cumplir con ellos.



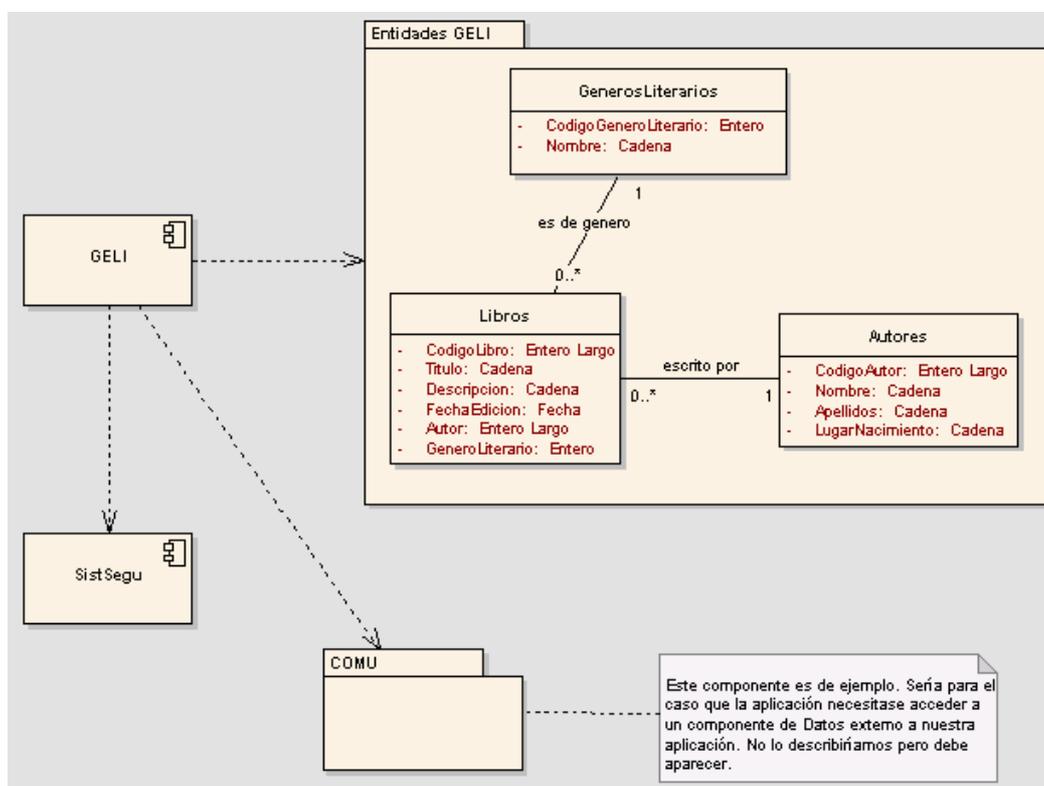
Este modelo será útil para después realizar el análisis de los casos de uso puesto que se puede ver, a alto nivel, las actividades que se pueden realizar en el sistema. Cada una de estas actividades darán lugar a uno o varios casos de uso.

En este caso, puesto que la aplicación es tan sencilla, no sería necesario hacer este diagrama, sería suficiente con el diagrama de casos de uso.

Cada una de estas actividades engloban acciones manuales, o un conjunto de casos de uso si se trata de actividades automáticas.

4 Modelo del dominio

En este punto debemos identificar todas las entidades del sistema que vamos a desarrollar y las relaciones que se dan entre ellas. Para cada una de estas entidades vamos a indicar los objetos que van a manejar para cumplir la funcionalidad necesaria. Podría obtenerse un diagrama en el que las entidades, inicialmente, no tuviesen atributos (en nuestro caso, estos atributos aparecen por la propia definición del sistema). A medida que se vaya realizando el análisis se irán completando estas entidades con sus atributos y métodos. Además se debe mostrar el entorno en el que se encuentra la aplicación, con que sistemas interactúa, componentes a los que accede, etc. En nuestro caso, estamos trabajando con tres entidades, libros, géneros literarios y autores, y necesitaremos de un sistema de seguridad para comprobar la identidad de los usuarios. Podría ser necesario acceder a objetos de BD comunes. En ese caso lo reflejaríamos como un paquete vacío, puesto que el contenido queda fuera de nuestro proyecto. En la siguiente figura se puede ver el modelo planteado.

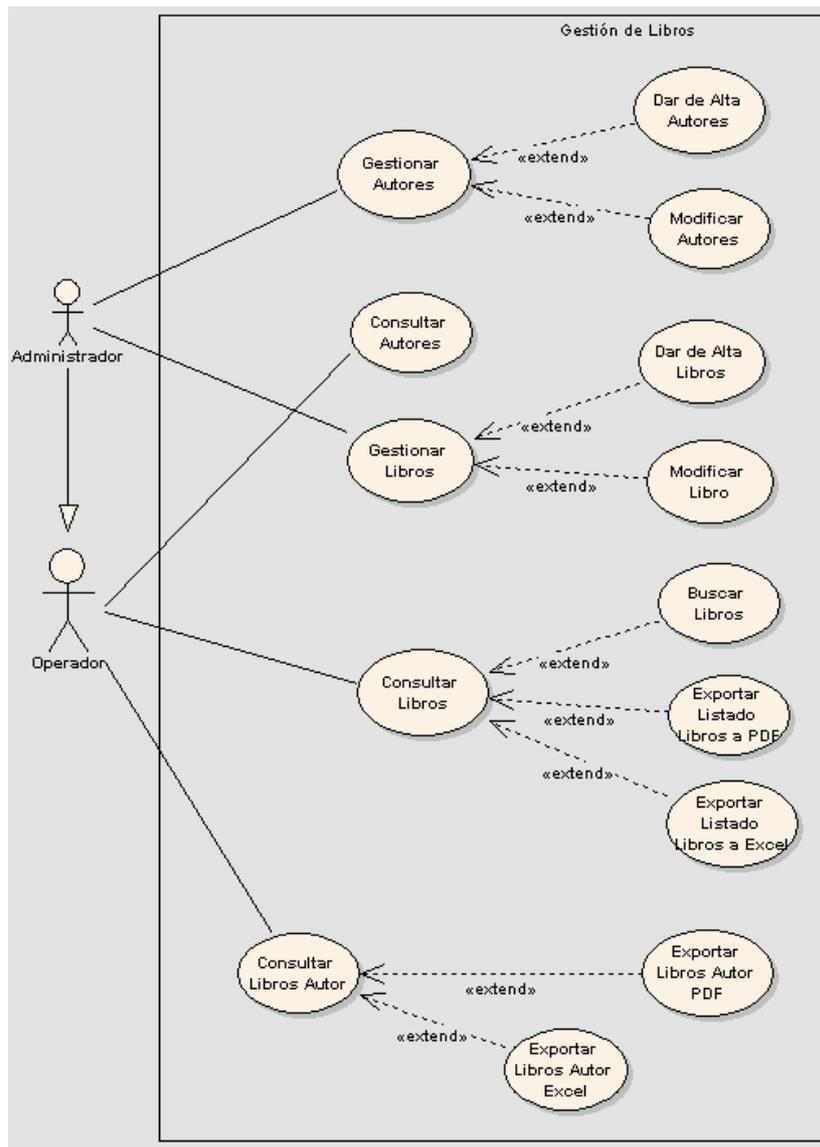


5 Entorno tecnológico

En este caso no es necesario desarrollar este apartado puesto que con el modelo del dominio ya tenemos la visión de lo que debemos resolver. Este apartado podrá ser útil para sistemas complejos en los que se pueda ver la estructura final sobre la que se va a desarrollar la aplicación. En este caso, además, por tratarse de una aplicación J2EE de la Junta de Castilla y León se puede evitar incluso siendo una aplicación mas compleja, puesto que siempre se va a utilizar el mismo entorno tecnológico.

6 Modelo de casos de uso

El siguiente apartado que vamos a abordar es la definición de los diferentes casos de uso identificados a partir de los requisitos. Si en esta definición encontramos algún requisito que no habíamos identificado en el paso 1, podemos redefinir esos requisitos para reflejar los cambios. Para nuestra aplicación hemos identificado los siguientes casos de uso.

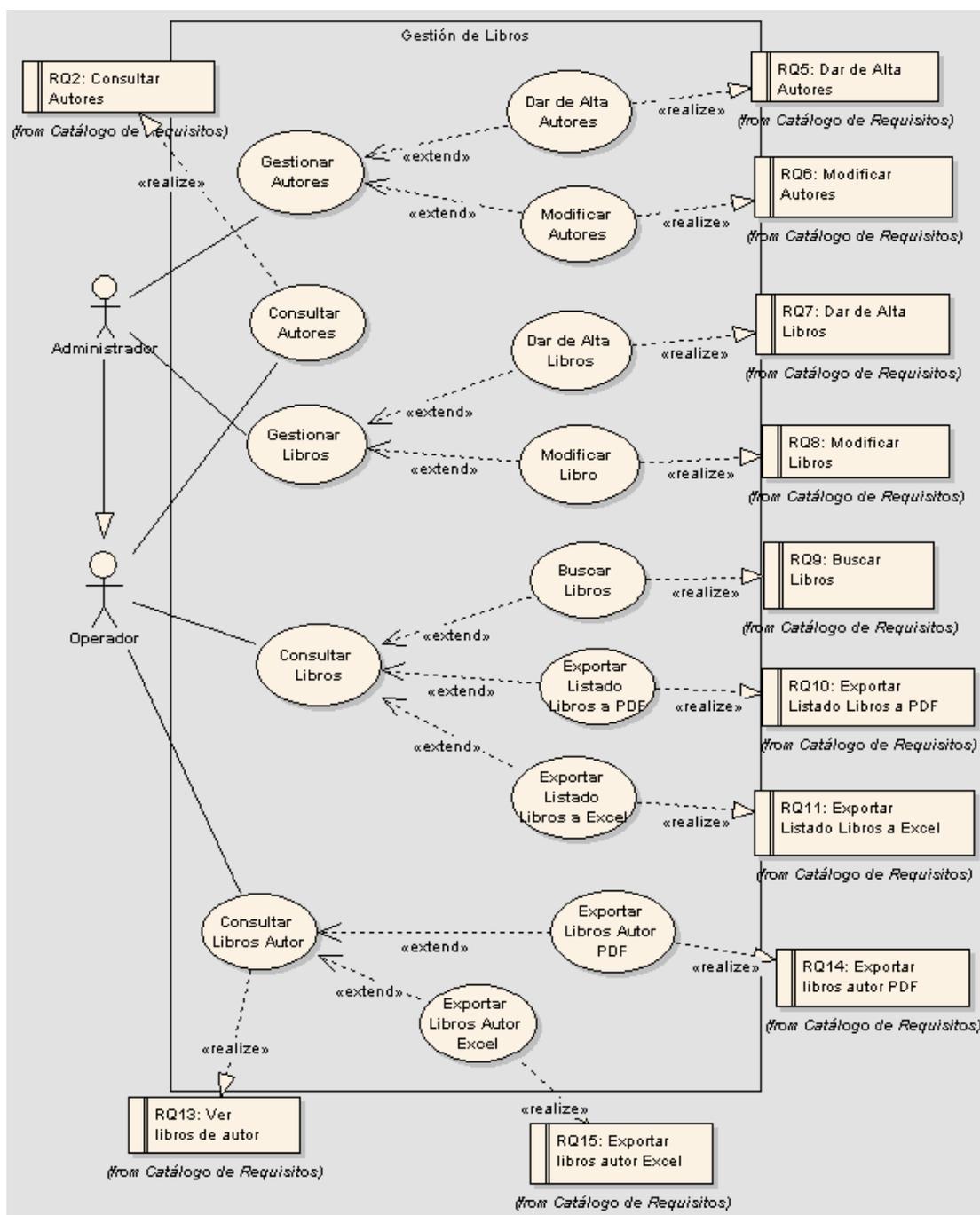


Este punto va a ser muy importante para posteriormente construir los **cuadernos de carga**, puesto que **habrá uno por cada caso de uso** que definamos aquí.

Dada la sencillez de nuestra aplicación, y que el número de casos de uso que hemos identificado no es muy grande, hemos puesto todos dentro del mismo diagrama. Si este número hubiese sido mayor, habría que haber descompuesto este diagrama en paquetes de forma que dentro de cada paquete se tuviesen divididos los casos de uso. Con esto estaríamos identificando los subsistemas que se pueden encontrar en el sistema.

Para cada uno de estos casos de uso se pueden indicar casos de prueba y realizar las estimaciones que se crean oportunas, tiempo de desarrollo, coste,....

A continuación vamos a mostrar un diagrama en el que se muestra, para cada caso de uso a que requisito hace referencia.



7 Análisis de los casos de uso, diagrama de clases de análisis y detalle de pantallas

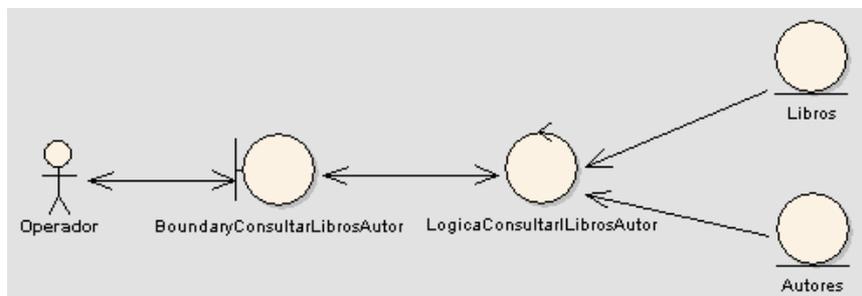
En este apartado hemos incluido tres puntos que en la figura inicial se pueden ver como independientes. Esto es así porque pueden no ser necesarios si se trata de una aplicación sencilla. Se puede pasar directamente a las realizaciones de diseño y clases de diseño. En el caso de las pantallas, no es necesario puesto que en los cuadernos de carga vamos a detallar todas las pantallas necesarias.

Si el usuario al que va dirigido este documento está próximo al programador, el desarrollo de los puntos 8.1 y 8.2 puntos puede generar errores de concepto, puesto que se puede incluir en el análisis cuestiones que deberían ser de diseño.

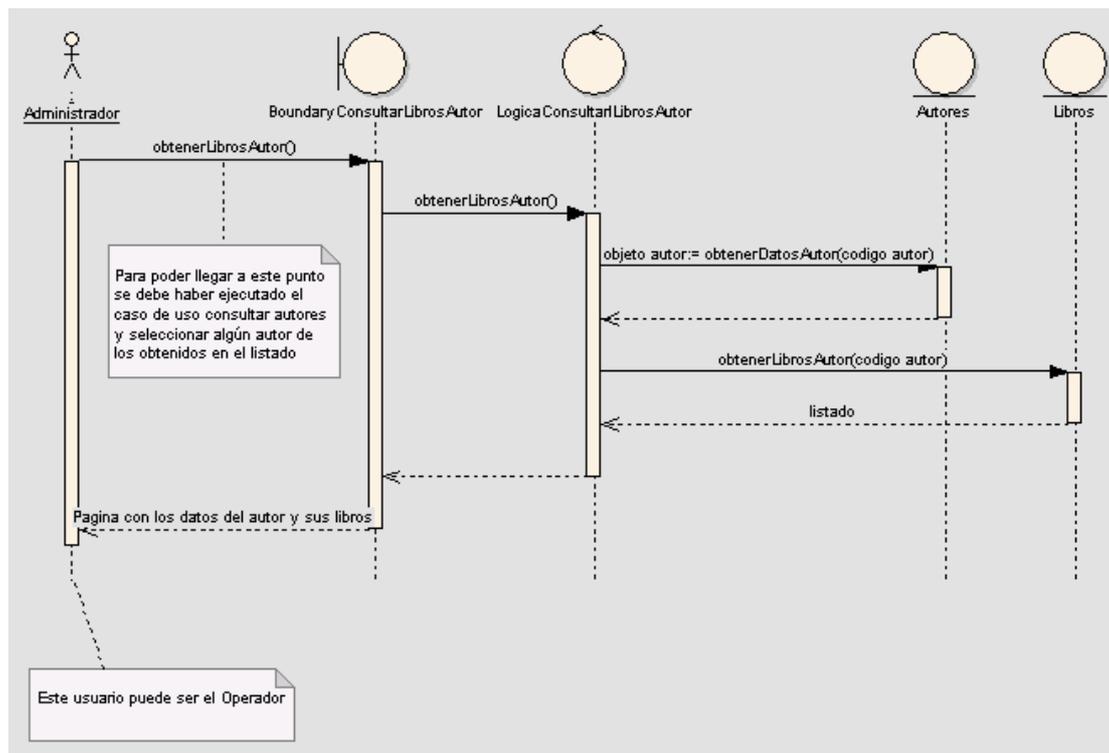
7.1 Análisis de los casos de uso

En este apartado vamos a detallar los casos de uso identificados en el punto 6 de esta documentación. Estará dentro de la carpeta “Realizaciones de análisis”. **No será necesario hacerlo para todos, sólo aquellos para los que se vea complejidad y sea necesario aclararlo mediante diagramas de secuencia.** En nuestro caso, puesto que todos los casos de uso son muy sencillos, no sería necesario describirlos todos, hemos hecho alguno para que se vea el resultado que se espera obtener. Vamos a ver el análisis para el caso de uso “Consultar libros autor”.

Vemos un diagrama en el que se ve la comunicación entre los diferentes elementos necesarios en el caso de uso



Y un diagrama de secuencia en el que se ven los mensajes que se pasan entre los diferentes elementos para conseguir la funcionalidad del caso de uso.



Una vez terminado el análisis de los casos de uso, habremos completado las entidades identificadas en modelo del dominio, y añadido los métodos a los que podremos invocar para obtener la funcionalidad que puede proporcionar cada entidad.

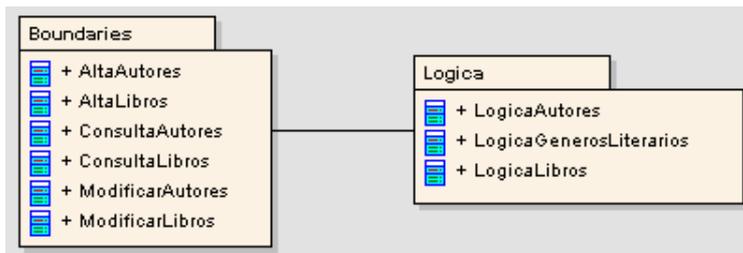
No será necesario hacerlo para todos, sólo aquellos para los que se vea complejidad.

Este punto, junto con los dos anteriores van a ser muy importantes en la definición de los cuadernos de carga puesto que se está identificando la secuencia de acciones que se llevan a cabo en cada caso de uso.

7.2 Diagrama de clases de análisis

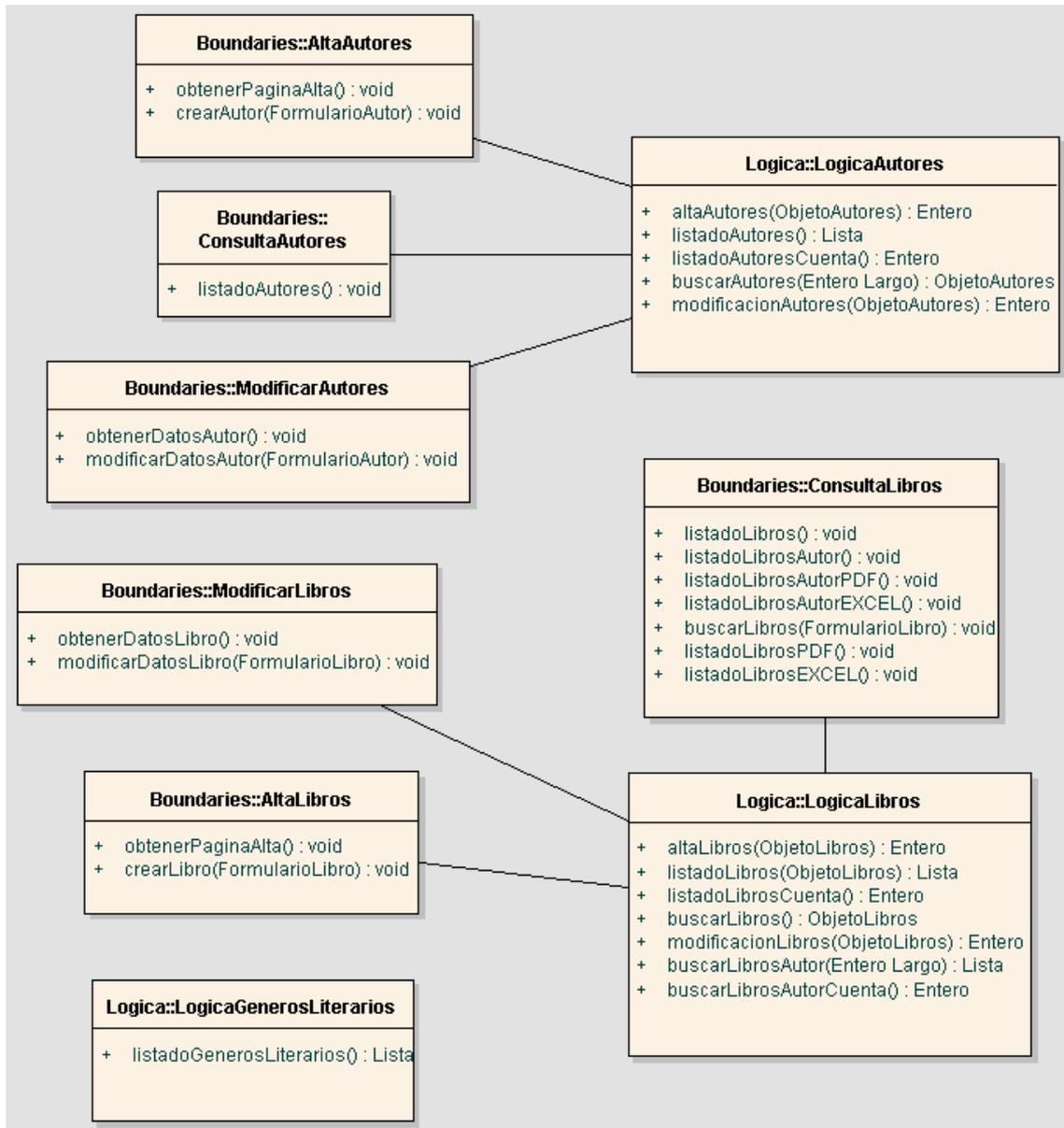
En este diagrama vamos a completar el modelo del dominio con la funcionalidad de cada entidad. Como hemos dicho en el punto anterior, ya habremos identificado parte de esas funcionalidades, y habremos identificado las nuevas entidades que vamos a necesitar.

En este paso, atendiendo a los diagramas de secuencia que hemos establecido, vemos que las entidades iniciales se han dividido en otras más pequeñas, a través de las que se va a conseguir la funcionalidad esperada. A continuación podemos ver el diagrama que hemos propuesto



Como puede verse, no todos los objetos Boundary y Control, tienen reflejo directo en las clases de análisis. Hemos pensado que los objetos BoundaryConsultarLibrosAutor y ControlConsultarLibrosAutor, se pueden incluir en la clase ConsultaLibros, donde se incluirán todos los boundaries que tienen que ver con las consultas de libros.

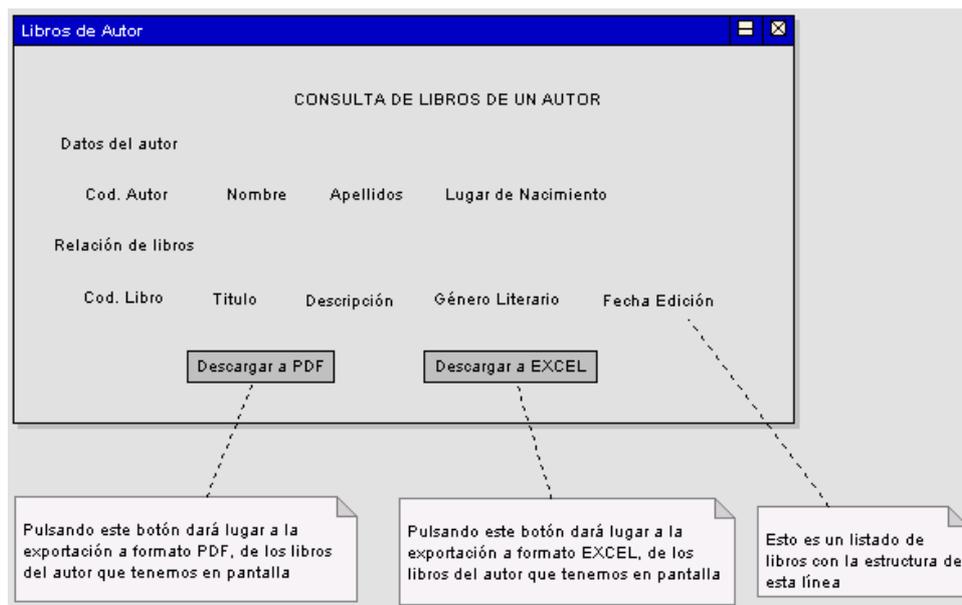
En este diagrama lo hemos agrupado por paquetes, de forma que se vean las clases que corresponden a los objetos boundary que hemos identificado en el análisis de los casos de uso, y por otro lado los controles, de forma que están relacionados entre ellos. En la siguiente imagen ya vemos la relación entre las clases de cada paquete.



Ya se va empezando a dar una visión general de todas las entidades necesarias en el sistema, que se plasmarán definitivamente en el diagrama de clases de diseño que veremos en el apartado 9 de esta documentación.

7.3 Detalle de pantallas

El último apartado que vamos a desarrollar dentro de la fase de análisis será la descripción de las pantallas que se proponen para la aplicación. Hemos creado un paquete dentro del diagrama por cada caso de uso, y con el mismo nombre para poder identificarlas de forma rápida. Vamos a ver la pantalla propuesta para el caso de uso que hemos descrito anteriormente en el diagrama de secuencia, “Consultar libros autor”. A esta pantalla llegaremos a partir de la consulta de todos los autores, pulsando sobre cualquiera de los autores que allí se muestren.

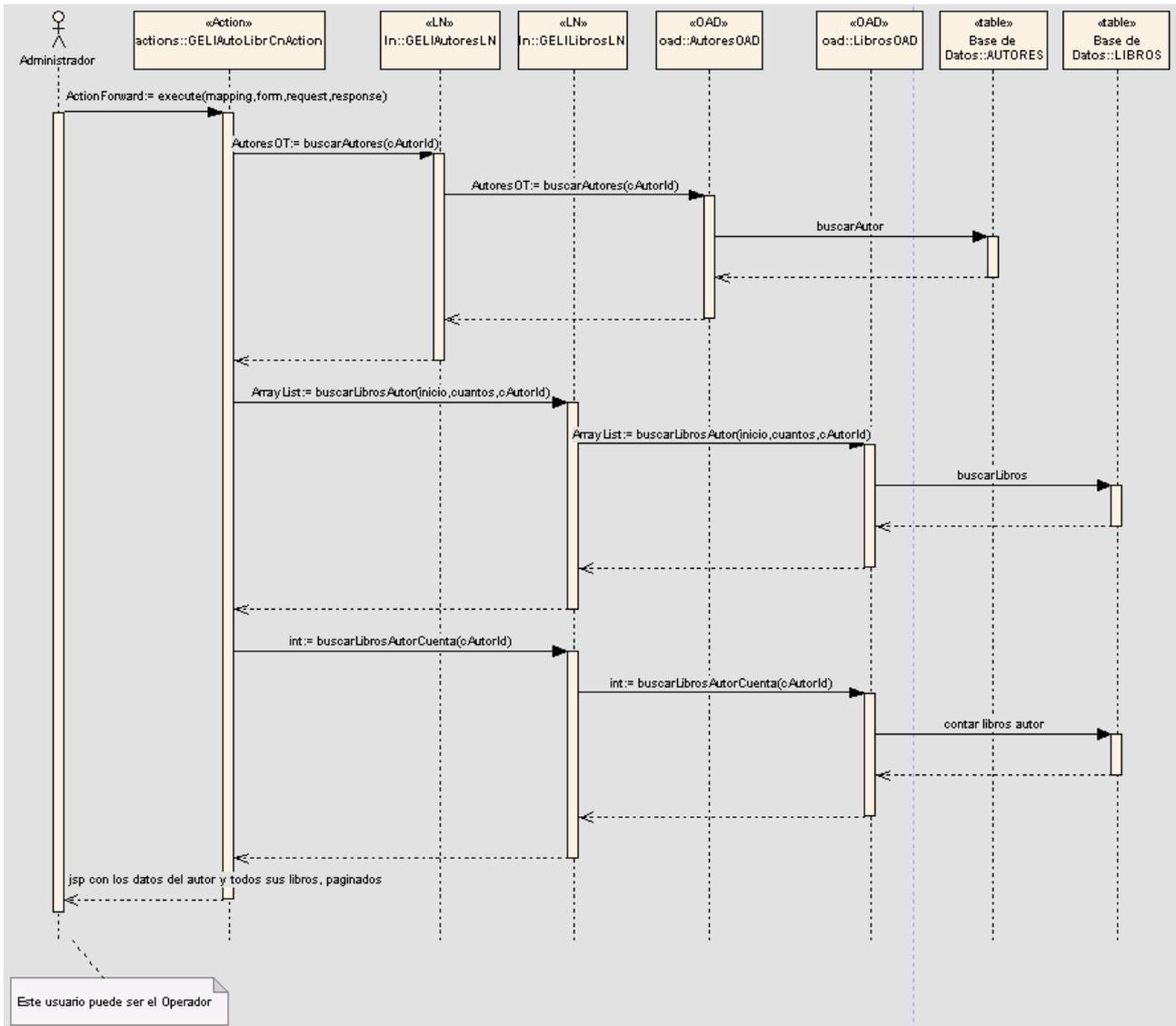


En la pantalla se pueden ver los datos mostrar, los botones de que se dispone. Para cada uno de ellos se muestra el caso de uso al que dan lugar en caso de pulsar sobre ellos.

La descripción de cada una de estas pantallas dependerá del tipo de usuario a quien vaya dirigida. Así, si el usuario es de nivel 1 ó 3, habrá que describirla de forma detallada para que no haya ninguna confusión, mientras que si es de nivel 2 se podrán dejar algunos aspectos de la pantalla para que el usuario decida como hacerlo.

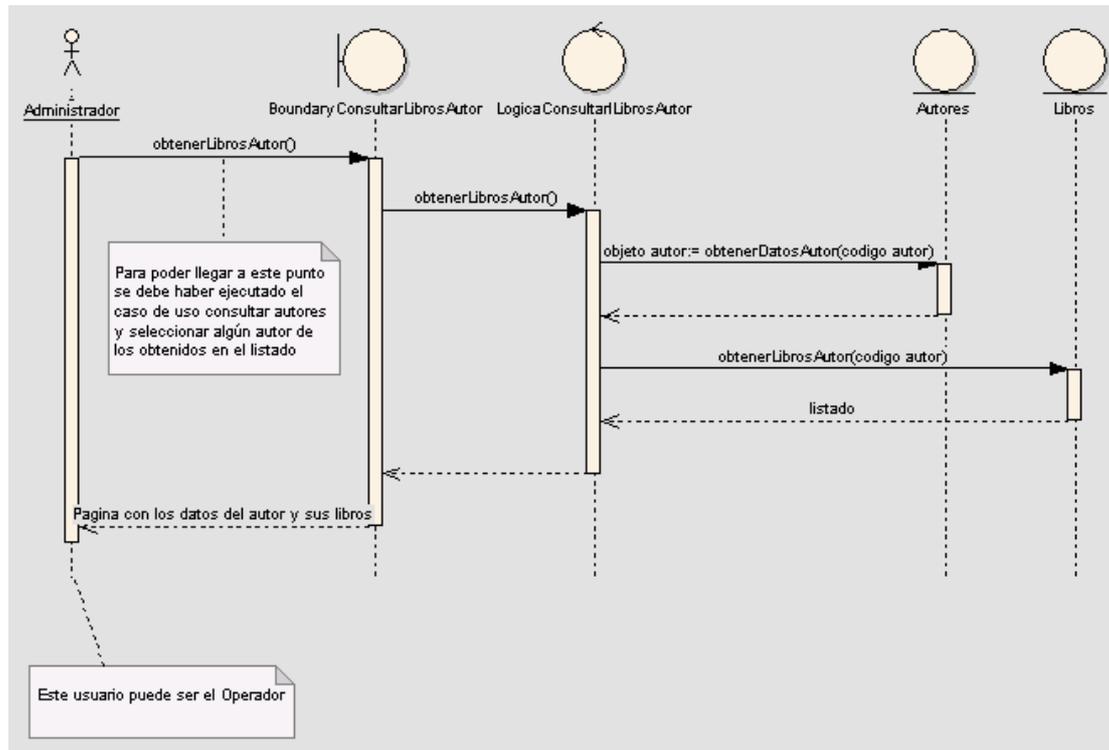
8 Realizaciones de diseño

Este será el **primer apartado del diseño** que vamos a construir. En este apartado vamos a profundizar en el detalle de los casos de uso, mostrando los detalles de cómo realizar cada uno de ellos, los métodos que se van a utilizar y las entidades que intervienen. Vamos a ver como queda el caso de uso “Consultar libros autor” que presentamos en la fase de análisis.



Igual que ocurría en análisis, **no vamos a generar documentación para cada caso de uso, sólo para aquellos que se vea cierta complejidad.**

Vamos a ver el paso del análisis al diseño, y la relación que hemos establecido entre objetos de análisis y diseño. Para ello vamos partir de este caso de uso en análisis



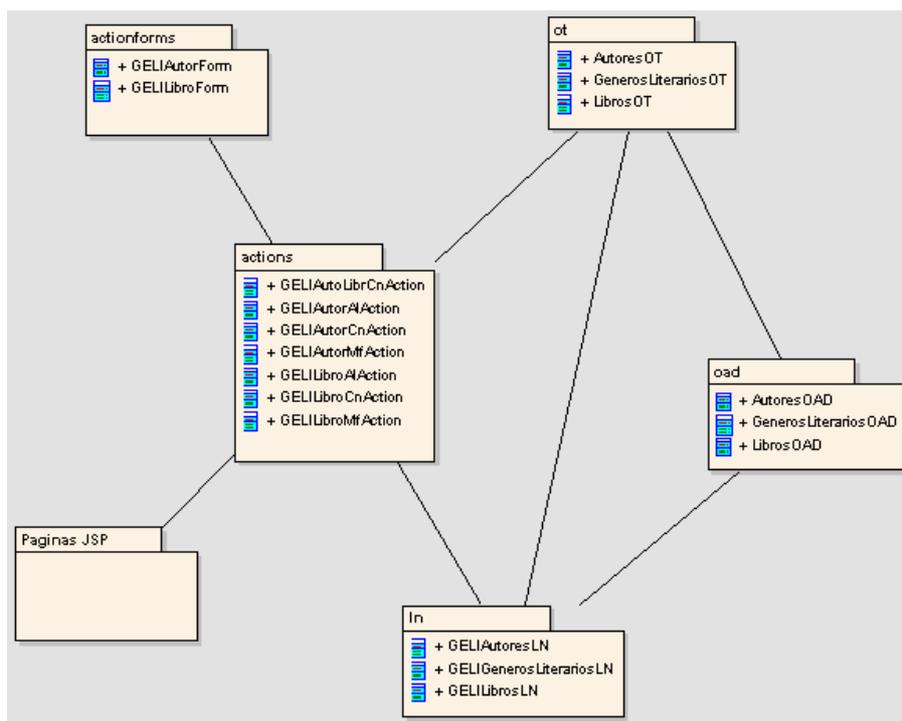
Lo que en análisis llamamos boundary, pasan a ser los actions y las jsps en diseño. Los objetos control en análisis (lógica) pasarán a ser las clases de la lógica de negocio (LN). Finalmente los objetos entidad pasarán a convertirse en las clases acceso a los datos y las tablas en las que se encuentran esos datos.

9 Clases de diseño

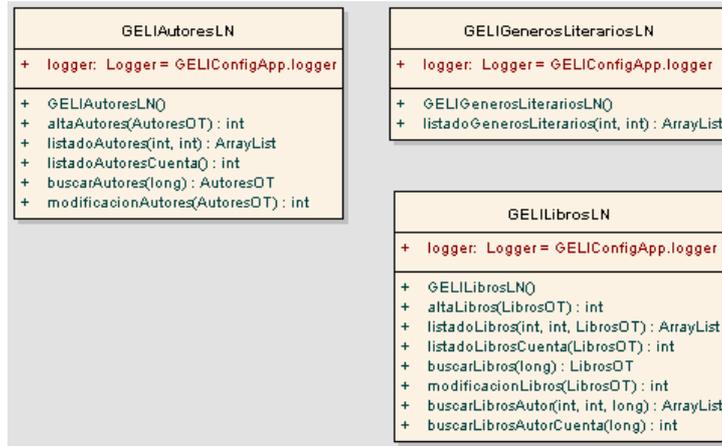
A partir de las clases de análisis que hemos identificado en el apartado 7, vamos a definir las clases que se van a desarrollar en la fase de construcción. En este momento ya se van a indicar los nombres definitivos de las clases, métodos y atributos que van a tener, pensando ya en el lenguaje que se va a utilizar para la construcción, y el entorno tecnológico con el que se va a desarrollar. En nuestro caso vamos a utilizar el estándar de la Junta de Castilla y León para el desarrollo de aplicaciones J2EE, que se basan en un framework propio apoyado en struts.

Este apartado será necesario desarrollarlo dependiendo del sistema que se esté construyendo y sobre todo, de los desarrolladores a quienes va dirigido. Si la aplicación es sencilla y hay pocos desarrolladores, pueden ellos mismos decidir los nombres de clases y métodos con los que van a implementar las funcionalidades del sistema. Si vamos a tener diferentes grupos de desarrollo que pueden necesitar componentes que van a desarrollar otros grupos, será necesario establecer todas las clases, atributos y métodos, de forma que se establezca el interfaz entre los diferentes grupos. Una vez definidas todas las clases, generar este diagrama es sencillo puesto que bastará con arrastrar las clases y paquetes al diagrama.

En nuestro caso, aun no siendo necesario, hemos generado todas las clases y métodos, y las agrupado en paquetes, de forma que las relaciones que se muestran se producen entre los diferentes paquetes, y no entre clases del mismo paquete.



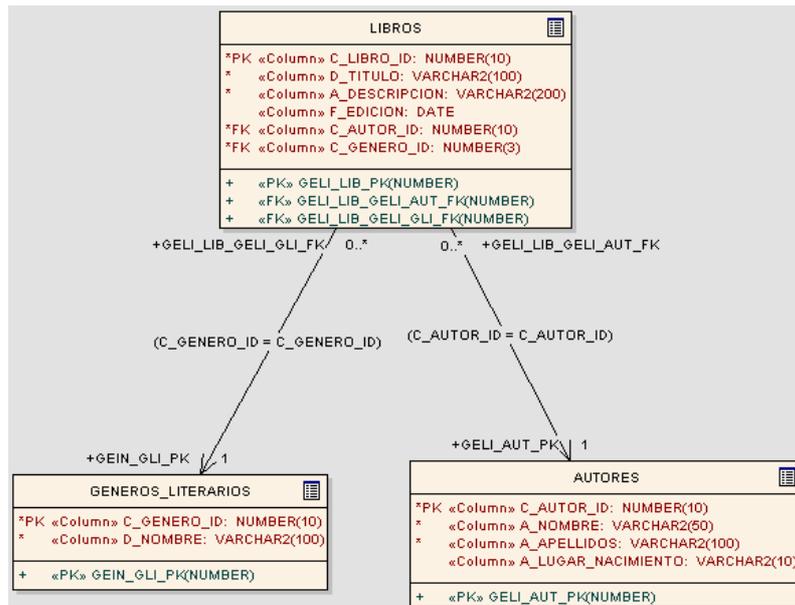
Vamos a ver las clases del paquete de la lógica de negocio.



En este punto se ha pasado de entidades a clases java, utilizando para su nombrado los estándares marcados por JCyL.

10 Base de datos

Finalmente, puesto que la aplicación debe mantener una serie de datos sobre libros, autores y géneros literarios, relacionados entre si, hemos optado por mantenerlos en una BD relacional. Es necesario definir la estructura de tablas y la relación entre ellas, de forma que se mantengan los datos necesarios para la aplicación. A continuación se puede ver la propuesta para nuestra aplicación



Si se observa la figura del apartado 1, en el panel derecho, se pueden ver todos los productos que se podrían generar, pero dada la sencillez de nuestro sistema, no será necesario realizar ninguno más.

11 Cuadernos de carga

Entramos en el **apartado objetivo de este documento**. Una vez finalizado el diseño, ya estamos en disposición de generar los cuadernos de carga que describen cada uno de los casos de uso y la forma de implementarlo. Estos cuadernos se pasarán a los desarrolladores de forma que el resultado obtenido sea el esperado.

Para conseguir esto, en función del nivel al que pertenezca el usuario desarrollador, habrá que describirlo de una forma o de otra. Utilizaremos los documentos que hemos ido generando en apartados anteriores para su descripción.

Vamos a generar el cuaderno de carga para un caso de uso sencillo, “Dar de Alta Libros”, y veremos que apartados debe contener y el grado de detalle con el que se deben describir.

11.1 Descripción

Para comenzar con la descripción del caso de uso, se indicará información general acerca de él, como el autor, nombre del caso de uso, fecha de creación, actores que pueden acceder a él, contexto en el que se puede utilizar (por ejemplo si es necesario que los usuarios estén validados, o puede acceder cualquiera... condiciones que deben cumplirse para ejecutar el caso de uso), forma de invocarlo, etc.

A continuación se describirá la secuencia de pasos que se deben seguir para obtener la funcionalidad. Esta será la secuencia de pasos normal. Para cada uno de estos pasos se pueden definir alternativas, como por ejemplo, si se produce algún error, que se debe hacer en ese caso, o que sucede al pulsar sobre algún botón de la pantalla.

Una vez explicado el funcionamiento, se puede plasmar esta definición en un diagrama, de forma que gráficamente se vea la secuencia de acciones que se han descrito anteriormente, y la secuencia de pantallas que se van a mostrar al usuario. Puesto que hemos generado toda la documentación del sistema, ya tendremos las acciones, objetos y pantallas que puedan ser necesarias para el caso de uso, y por tanto, será sencillo representarlo.

A partir de la descripción detallada y de su representación gráfica, vamos a indicar todos los objetos que van a ser necesarios dentro del caso de uso, agrupándolos por funcionalidad. Así, para nuestro caso, indicaremos las acciones, formularios, jsps, objetos de lógica de negocio, objetos de acceso a datos y objetos de transferencia. Será un simple listado de objetos, y si es necesario, se indicarán los métodos dentro de cada objeto que se deben utilizar.

Finalizado el paso anterior, continuaremos con la descripción detallada de cada uno de los objetos indicados en él. **El nivel de detalle al que se llegará en esta descripción va a depender del nivel al que pertenezca el usuario.**

Para describir los casos de uso, se han definido tres plantillas de acuerdo con el detalle al que se debe llegar en su definición, siempre en función del usuario al que vaya dirigido. En cada una de ellas se ha desarrollado el caso de uso teniendo en cuenta ese nivel, pudiendo servir como ejemplo para otros casos de uso.

Pulsando en los enlaces que se muestran a continuación se puede obtener la plantilla de ejemplo desarrollada para nuestro caso de uso, y para cada uno de los usuarios que hemos considerado al comienzo de este documento.

[Usuario Nivel 1](#)

[Usuario Nivel 2](#)

[Usuario Nivel 3](#)

Observando estas plantillas, y viendo el nivel de detalle con el que se crea el cuaderno de carga, se puede incrementar el tiempo de desarrollo de las aplicaciones, pero se va a reducir de forma considerable su coste de mantenimiento.

11.2 Diferencias en el nivel de detalle en las plantillas

Una vez presentadas las diferentes plantillas con el contenido para nuestro caso de ejemplo, vamos a comentar las diferencias entre cada una de ellas.

Las plantillas del nivel 1 y 3 son iguales puesto que, aun cuando los usuarios externos puedan ser avanzados, y a nivel interno no necesiten una documentación tan detallada, es necesario que la documentación del cuaderno de carga sea amplia para que en el periodo de mantenimiento cualquier persona interna a la organización de JCyL pueda realizar los cambios necesarios con el menor coste. Esto mismo se podría aplicar a la plantilla del usuario de nivel 2, puesto que en un principio él será el desarrollador, pero en el periodo de mantenimiento podría ser un usuario de nivel 1 quien tendría que realizar los cambios y en este caso la descripción debería ampliarse.

Los dos primeros apartados son similares, incluso iguales, puesto que se define detalladamente el caso de uso. En el caso de nivel 2 puede obviarse alguna cuestión, como puede ser indicar que la validación se haga en javascript. Si esta descripción se hace detallada, facilitará que el resto de los puntos puedan ser menos extensos ya que el usuario, simplemente con esta descripción, se hace una idea clara de lo que debe desarrollar.

Para un usuario avanzado no es necesario detallar todos aquellos caminos alternativos que sean comunes a todas las aplicaciones. En nuestro caso, no es necesario detallar los caminos 1.a, 9.a y 9.b puesto que son validaciones normales en cualquier aplicación de este tipo.

En el siguiente apartado, la representación gráfica, también puede verse que es menos extensa, sobre todo en cuanto a las notas aclarativas que se han incluido, ya que un usuario avanzado, con un simple vistazo reconoce todo lo necesario para implementar el caso de uso.

A continuación hemos identificado los elementos necesarios para realizar el caso de uso. Para un usuario avanzado podría no ser necesario puesto que a partir de la definición y viendo el diagrama, será capaz de identificarlos de forma rápida y sencilla. De todas formas, se pueden dejar para que todos los desarrolladores sigan la misma estructura, y el nombrado de objetos.

Finalmente se describen en detalle cada uno de estos elementos. En el caso del usuario inexperto se entra muy a fondo en cada uno de los elementos de forma que no se deja lugar a dudas en cuanto a la forma de implementar cada uno de los objetos. En el caso de los objetos de acceso a datos, se detalla incluso la sentencia SQL que se debe ejecutar. Sin embargo, en el caso del usuario avanzado, esto no es necesario puesto que a partir de la funcionalidad esperada, será él quien decida la sentencia que va a ejecutar. En el caso de las páginas JSP, se puede ver que en el caso del usuario avanzado, no se han incluido las pantallas generales de mensajes y de error. Esto tampoco será necesario incluirlo en todos los casos de uso para el resto de usuarios puesto que, una vez indicado su funcionamiento en un caso, será común para todos los demás.