

Diseño

INDICE – Diseño

3	Diseño	117
3.1	Introducción	117
3.2	Patrones de diseño.....	118
3.2.1	Introducción	118
3.2.2	Patrón de diseño Modelo-Vista-Controlador (MVC)	118
3.2.3	División de la aplicación en paquetes.....	121
3.3	Diagramas de Interacción.....	123
3.3.1	Diagramas de Secuencia	123
3.3.2	Diagramas de colaboración.....	162
3.4	Diagramas Estáticos.....	168
3.4.1	Diagramas de clases para el Modelo (Acceso a los Datos).....	168
3.4.2	Diagrama de clases para el Modelo-Controlador.....	171
3.5	Diagramas de Actividad.....	175
3.6	Diagramas de estado	178
3.7	Diagramas de despliegue	179
3.8	Diseño de la base de datos	180
3.8.1	Introducción	180
3.8.2	Diseño de la estructura conceptual (Modelo Entidad-Relación) ..	180
3.8.3	Descripción de las Entidades	182
3.8.4	Descripción de las relaciones entre entidades.....	191
3.8.5	Diseño lógico (Paso a tablas y normalización).....	201
3.9	Bibliografía	215

3 Diseño

3.1 Introducción

Definición: *“Es el proceso por el cual se aplican distintos principios y técnicas con la finalidad de definir un producto con suficientes detalles que permita su realización”*

La fase de diseño pretende definir como se ha de construir un sistema que cumplirá ciertos criterios:

- Debe de satisfacer las especificaciones que se han definido en la fase de análisis, independientemente de que estas se han elaborado de forma informal o formal.
- Debe de ajustar el sistema que se ha de construir con las limitaciones impuestas por el entorno en el cual se va a implantar.

Estamos ante la primera etapa técnica en el proceso del desarrollo del software. Esta se basa en la construcción de un modelo que representa el software que se va a desarrollar y que es fundamental para la elaboración de software de calidad. También esta fase es clave, no sólo para crear software de calidad, si no de establecer los mecanismos que nos aseguren que realmente se están cumpliendo estos criterios de calidad. De igual modo un buen diseño nos permitirá la realización de un software altamente reutilizable, adaptable, ampliable y mejorará en gran medida el mantenimiento de este.

En el desarrollo de esta aplicación, se hará uso del paradigma de la programación orientada a objetos. Las principales características de esta tecnología son entre otras, la herencia, la alta abstracción, el encapsulamiento, la modularidad, la jerarquización, transparencia y el polimorfismo que hace de esta tecnología (POO) poseedora de grandes ventajas frente a la programación estructurada.

3.2 Patrones de diseño

3.2.1 Introducción

En los últimos años se está implantando la utilización de tecnologías de componentes, la más conocida JAVA y la más novedosa .NET, para el desarrollo de aplicaciones Web. Está muy generalizada la división de la arquitectura de la aplicación en tres capas:

- **Capa de presentación**
- **Capa de aplicación**
- **Capa de datos**

La **capa de presentación** se caracteriza por ser la encargada de interactuar con los distintos usuarios y que se corresponde con la Interfaz de Usuario.

En cambio, la **capa de aplicación** es la encargada de realizar la lógica de negocio, para lo cual, recibe de la capa de presentación las peticiones de los distintos usuarios y procede a llevarlas a cabo.

Por último, la **capa de datos**, se encarga de proporcionar a la capa de aplicación de acceso a los almacenes de la información que hay disponibles en el sistema.

Un diseño en el que se utilice este tipo de arquitectura, permitirá la creación de código modular que a su vez facilita la modificación de estos módulos sin necesidad de modificar otros.

3.2.2 Patrón de diseño Modelo-Vista-Controlador (MVC)

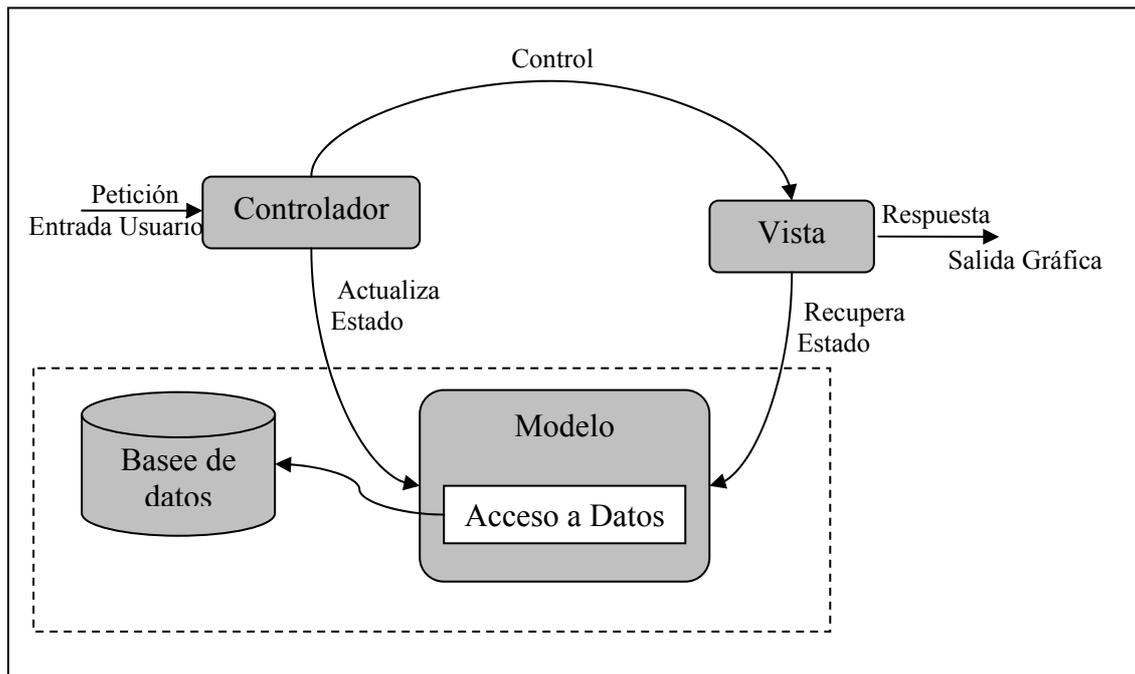
Se trata de un patrón de diseño específico de la capa de presentación, ya que el resto de capas se representa posteriormente gracias al modelo lógico del diseño. El principal objetivo de este patrón es la separación de la lógica de negocio frente a la presentación de la aplicación.

Este patrón descompone una aplicación interactiva en tres grandes bloques:

- El **modelo** contiene el control de la funcionalidad de la aplicación. Este encapsula el estado de la aplicación sin saber nada de las otras dos categorías: Vista y Controlador.
- La **Vista** proporciona la presentación del Modelo, representando el *look* o apariencia de la aplicación. La vista puede acceder a los métodos “*get()*” del Modelo para obtener información, pero no tiene acceso a los métodos “*set()*” para proporcionar información al Modelo, sino que las actualizaciones en el Modelo deben realizarse a través del Controlador. Además, la Vista será notificada cuando se produzca algún cambio en el Modelo.

- El **controlador** es quien reacciona a las acciones del usuario y es el encargado de crear y asignar valores al Modelo para su funcionamiento.

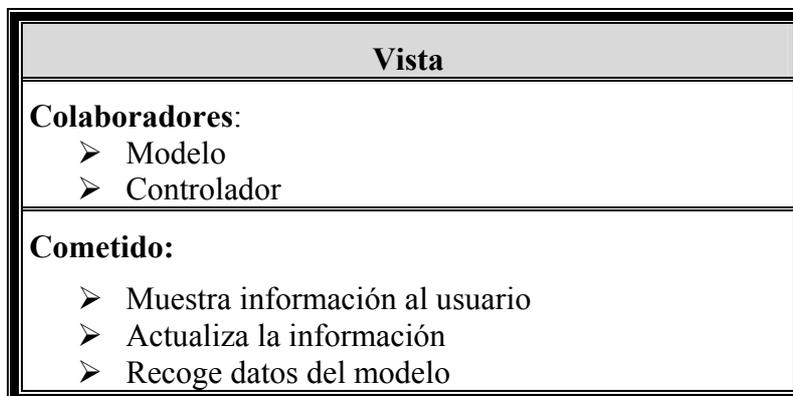
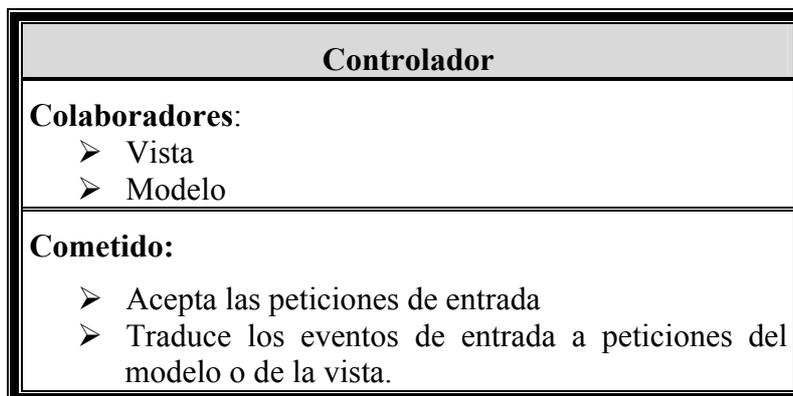
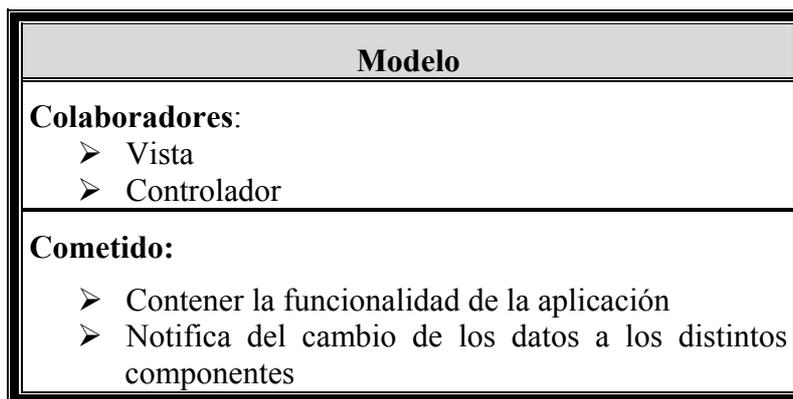
La diferencia que aporta la incorporación de la filosofía Web al patrón de diseño MVC es la dificultad en la implementación del Modelo de la notificación de cambios en su estado a la Vista, porque el entorno Web es un entorno sin estados; es decir, el navegador tendrá que realizar peticiones periódicas, o bien el Modelo informará periódicamente de su estado, haya o no cambiado, siendo la Vista quien decida si presenta la información o no, comparando los datos que recibe con la situación actual.



Otra diferencia importante respecto al patrón de diseño MVC clásico es que en el entorno Web, la Vista puede utilizar una tecnología diferente en su implementación respecto a la utilizada en la implementación del Modelo y el Controlador.

El navegador del usuario, en su petición HTTP, genera una solicitud que es atendida por el Controlador. Este analiza sus entradas y llama a los objetos correspondientes del Modelo. El Modelo se encarga de ejecutar la solicitud y generar los resultados que mostrarán posteriormente las Vistas.

Según el resultado que retorne el Modelo, el Controlador derivará la generación de la página interfaz a una o más JSPs (Vistas), que podrán consultar los Modelos con el fin de realizar su tarea adecuadamente.



Principales ventajas e inconvenientes

Ventajas:

- Múltiples vistas del mismo modelo
- Vistas sincronizadas
- Gran modularidad
- Independencia entre la vista-modelo-controlador
- La aplicación puede soportar diversos tipos de interfaz de usuario.
- Podemos realizar cambios de manera sencilla del Interfaz de la aplicación.

- Esta arquitectura es a la vez distribuida y escalable, soporta transacciones y todas las características que un servidor de aplicaciones puede ofrecer: pool de base de datos, clustering, etc.

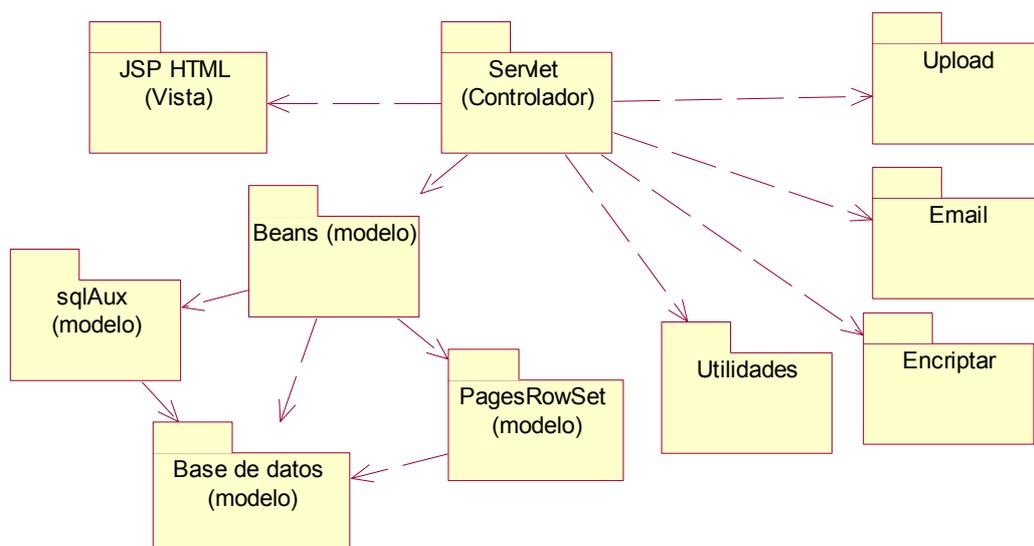
Inconvenientes:

- Complejidad creciente
- Si cambia la interfaz del modelo, hay que cambiar todas las vistas y controladores.
- Gran conexión entre el modelos y las vistas.
- Acoplamiento entre las vistas y el controlador al modelo.
- Acceso ineficiente a los dato en la vista. Pueden ser necesarias varias llamadas al modelo para actualizar los datos.
- Algunas herramientas de diseño de interfaces de usuario incorporan parte del procesamiento de eventos entrada. El controlador deja de ser necesario.
- El coste de tiempo adicional en la obtención de la respuesta por parte del servidor a una petición del cliente. Ello se debe a que es necesario un procesamiento intermedio de la petición; procesamiento que no se realiza cuando todo el código se encuentra en una sola página.

3.2.3 División de la aplicación en paquetes

Ya se ha visto en el punto anterior que se va a basar el diseño en este patrón MVC, vamos a representar, de forma esquemática, cada parte del modelo dentro de nuestra aplicación.

El modelo en nuestro caso se corresponderá con la base de datos y con las clases que controlan el acceso a esta, que será por mediación de javaBean. Las vistas en cambio serán loas páginas HTML o JSP que representarán la información que les proporciona el modelo y a su vez, esta proporcionará información al controlador sobre los eventos provocados por los usuarios. Por último el controlador está representado por servlets, que recogen información del usuario.



En este diagrama se observa distintas partes de la aplicación. En la parte superior tenemos los servlets de la aplicación que representan el controlador MVC. Son quienes reciben los datos del usuario y se los proporciona al modelo, encapsula la lógica de la aplicación. La base de datos es el modelo y los beans son los encargados de los accesos a esta, por tanto también pertenecen al modelo. Los beans se ayudarán de otros paquetes como `sqlAux` y `PagesRowSet` para poder acceder a los datos contenidos en la base de datos.

Tenemos una serie de paquetes que se encargan de proporcionar utilidades como el envío de correos electrónicos mediante SMTP, el envío de ficheros al servidor (upload) a través de peticiones http, otro que nos ayudará a encriptar y una serie de utilidades más generales.

3.3 Diagramas de Interacción

Estos diagramas son de dos tipos: de Secuencia y de Colaboración. Sirven para modelar aspectos dinámicos del sistema. Un diagrama de Interacción se usa para realizar una traza de la ejecución de un escenario, por lo que a cada escenario le corresponde un diagrama de Interacción [Booch 99].

Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Con todo esto, vamos a ver los diagramas de interacción (de secuencia y de colaboración) que hay en nuestro sistema.

3.3.1 Diagramas de Secuencia

Como ya vimos en el Análisis, los diagramas de secuencia modelan los casos de uso. Cada diagrama de secuencia contiene detalles de la implementación de un escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Captan una visión temporal del sistema. Su característica principal es que ponen especial énfasis en la ordenación temporal de los mensajes intercambiados por los elementos del sistema. Hay que tener en cuenta que se coloca a la izquierda el objeto que inicia la interacción, y el objeto subordinado a la derecha. También se utilizan en las fases de prueba para validar el código.

Esta vez, y a diferencia de los diagramas de secuencia incluidos en el análisis, que suponían una mera intuición de los objetos que serían necesarios para la implementación del escenario, veremos ahora los diagramas de secuencia que perfilarán los objetos finales que se emplearán en la implementación.

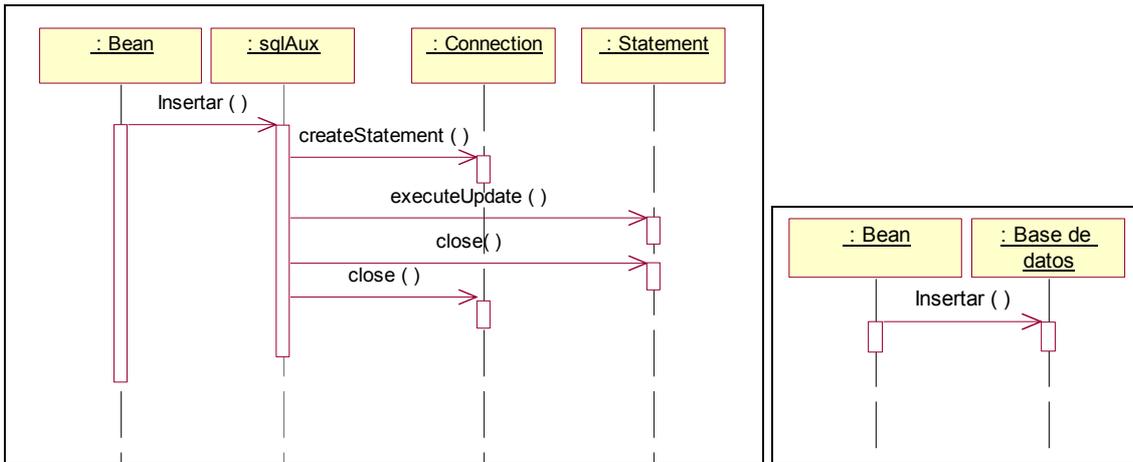
Diagramas gestión de la base de datos

La herramienta que se desarrolla se caracteriza por estar constantemente accediendo a la base de datos, por tanto, los diagramas de secuencia tendrán que reflejar estos accesos. Al producirse una repetición constante de información, los voy a abstraer. En esta abstracción voy a identificar estos procesos repetitivos de acceso y actualización de la base de datos y los voy a representarlo de forma más sencilla. Mediante esta abstracción pretendo realizar una documentación del diseño mucho más clara y entendible para el lector.

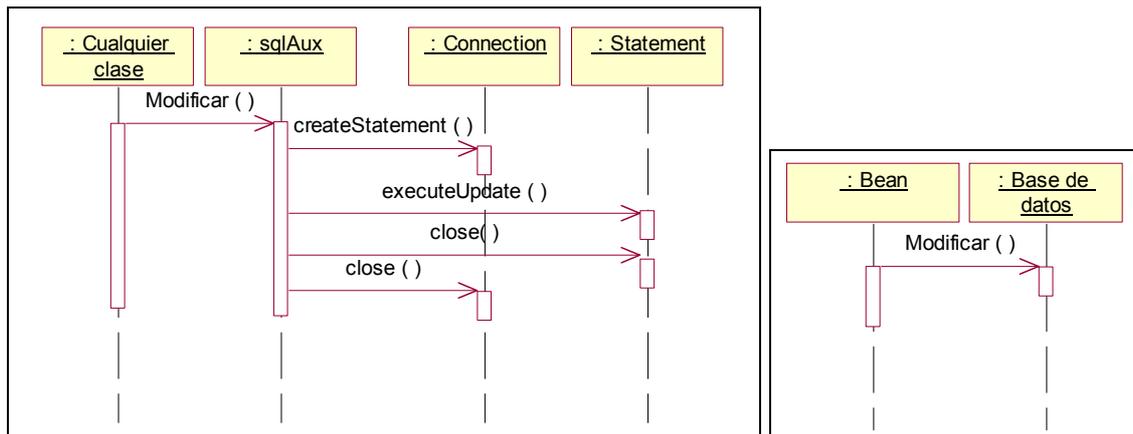
Como ya se ha comentado los beans son quienes acceden a la base de datos ayudándose para ello de auxSql o pagesRowSet. Donde auxSql es una serie de utilidades a la hora de trabajar con la base de datos y pagesRowSet, nos permite organizar el resultado de una consulta en distintas páginas.

Mostraré el diagrama original a la izquierda y como lo voy a representar a partir de ahora a la derecha.

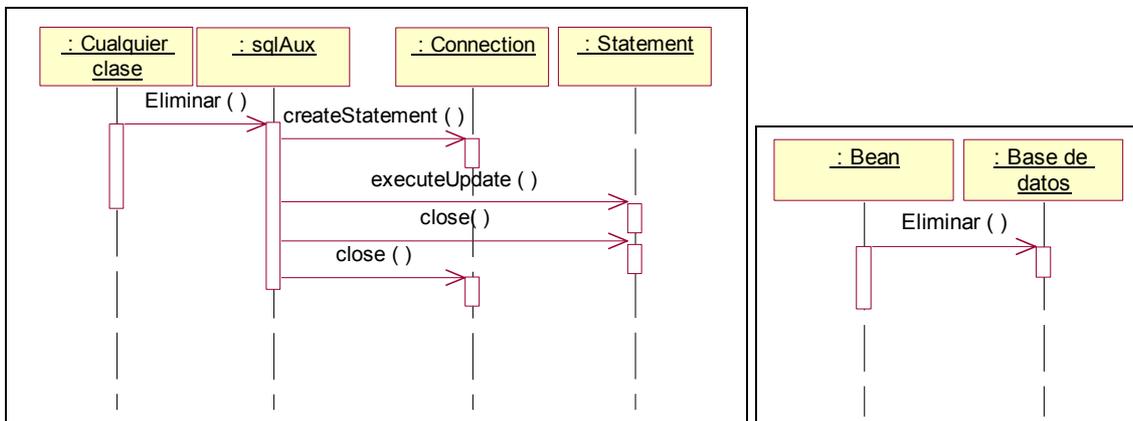
I) Insertar información en la base de datos



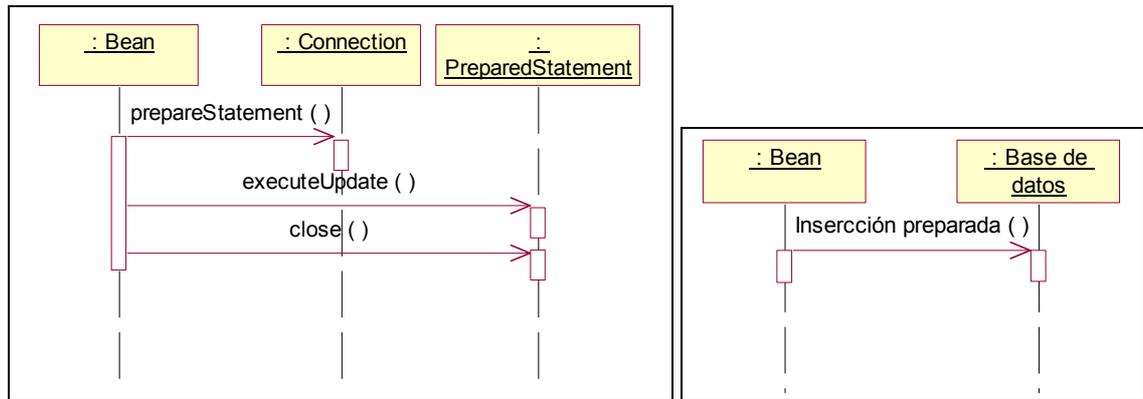
II) Modificar información en la base de datos



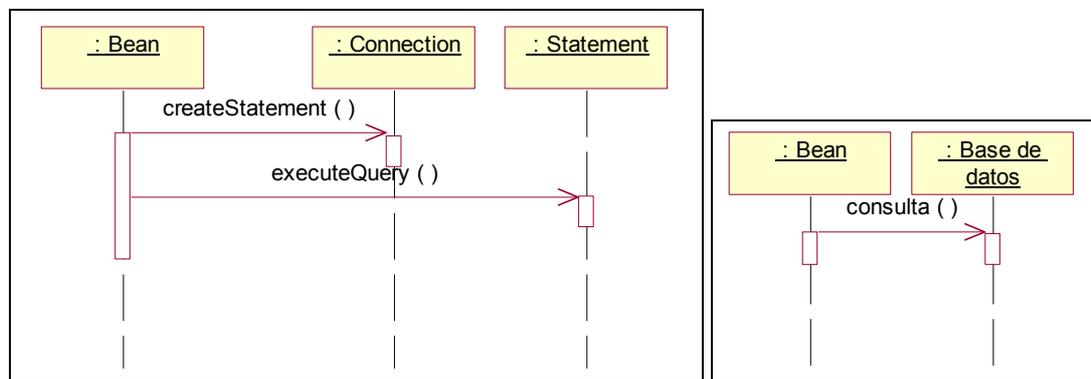
III) Borrar de la base de datos



IV) Inserción preparada en la base de datos

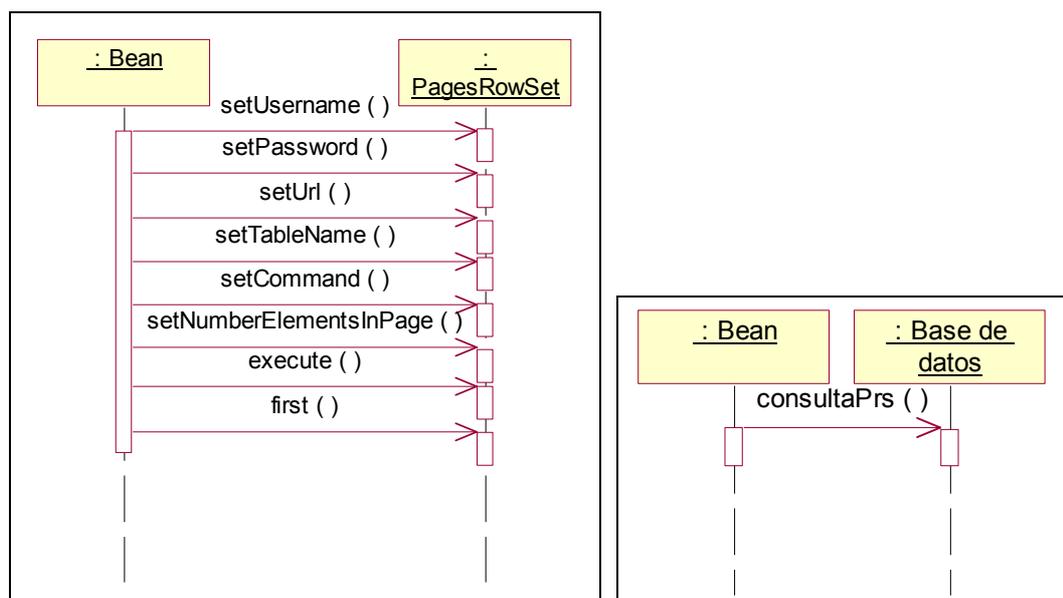


V) Consulta a la base de datos



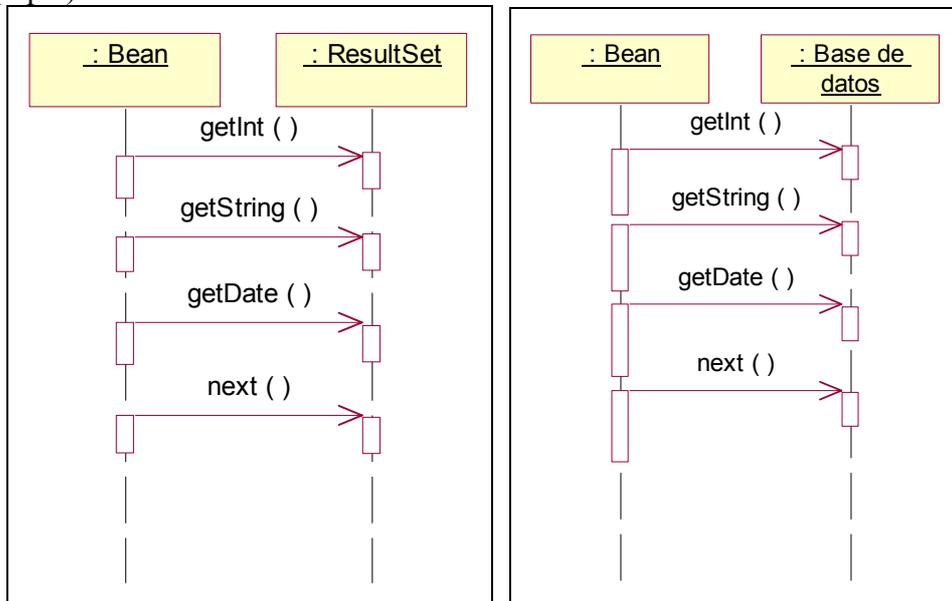
VI) Consulta PRS a la base de datos

En esta se nos devuelve la información en varias páginas

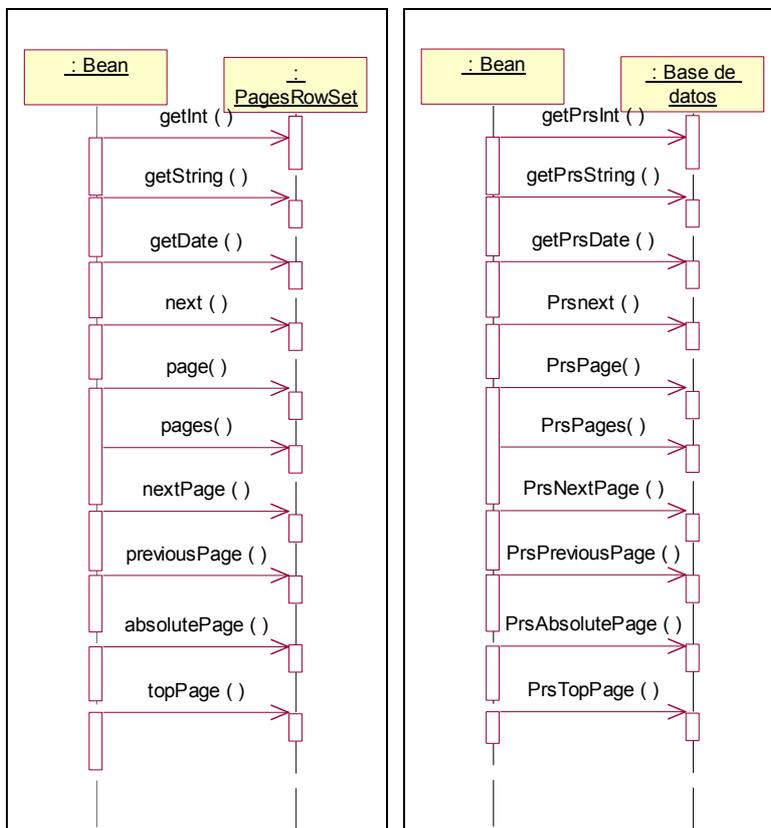


VI) Recuperación de la información y avance dentro de la base de datos

Integro en un solo diagrama todas las posibilidades que se utilizarán para sacar la información y a avanzar al siguiente elemento de información dentro de la base de datos (tupla).



VII) Recuperación de la información y avance dentro de la base de datos en una consulta PRS

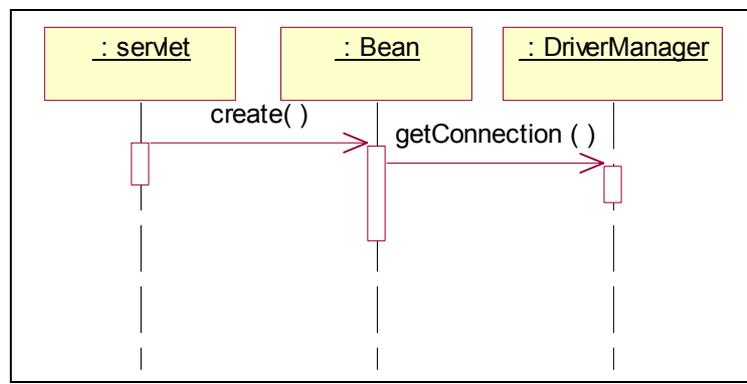


Diagramas inicialización

Como ya se ha comentado, el servlet se comportará como controlador y tendrá que comunicarse con el modelo (beans), por tanto tendrá que inicializar los beans que va a utilizar, aprovechará cuando se construye el servlet para instanciar los objetos beans, y estos cuando se crean aprovecharán para conectarse a la base de datos. De igual modo, cuando se destruya los servlets, estos mandarán a los beans instanciados que se desconecten de la base de datos. Las vistas podrán crear beans para hacer consultas, tomando el papel del servlet, también tendrá que desconectarlos cuando finalice la consulta.

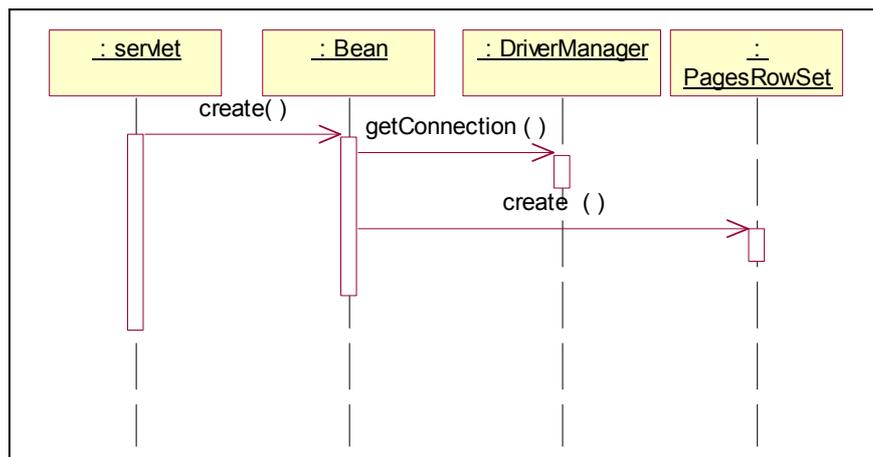
A partir de ahora cuando se cree o desconecte un bean sucederá los procesos aquí reflejados, pero que ignoraremos en el resto de diagramas.

VII) Inicialización beans

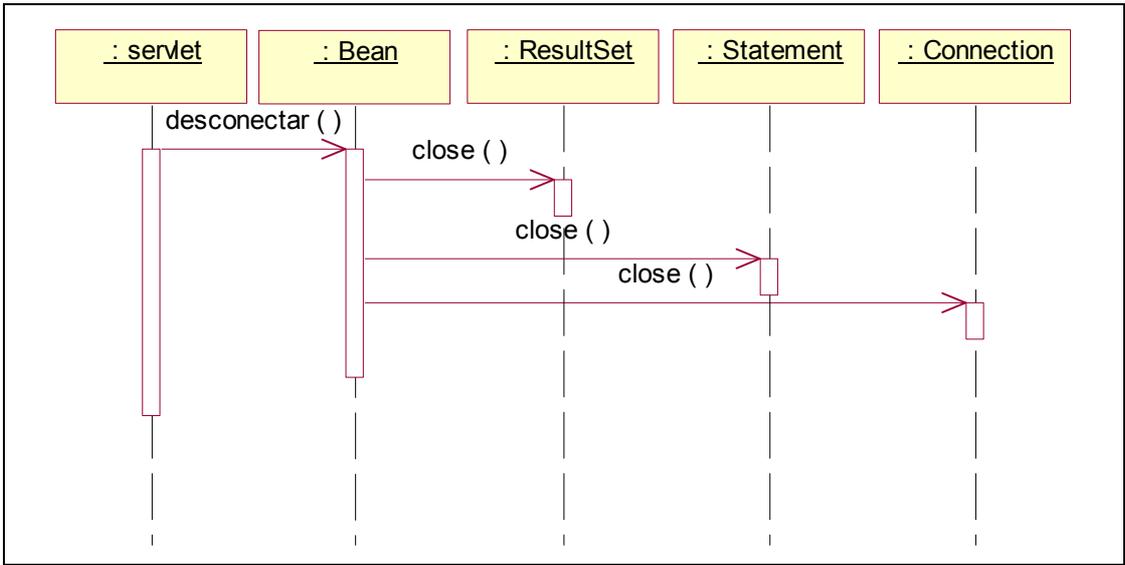


IIX) Inicialización beans con consultas almacenables en páginas

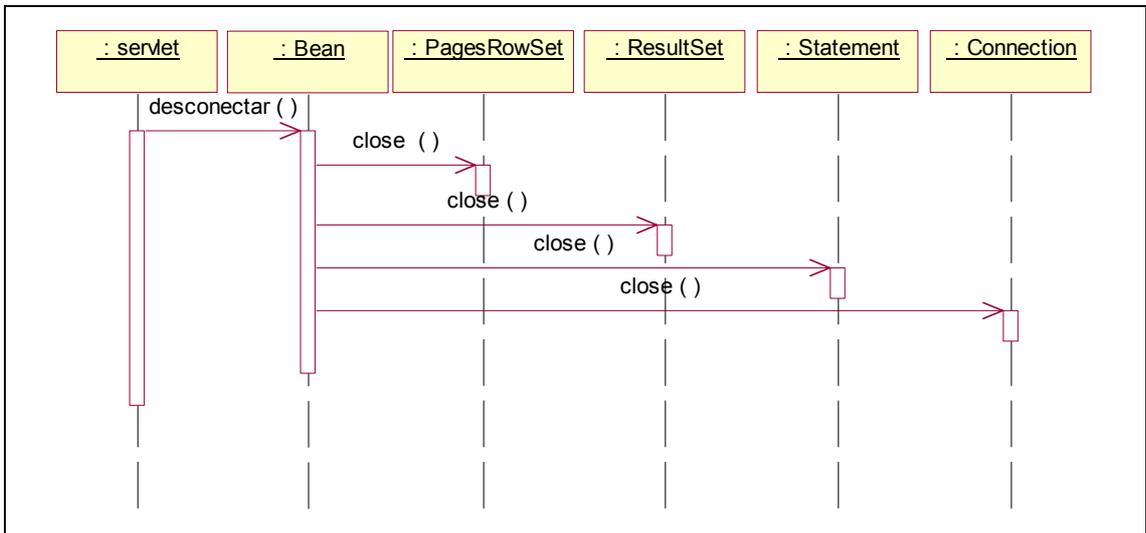
Estos tendrá un objeto PagesRowSet para almacenar el resultado de una consulta de este tipo.



IX) Desconexión de la base de datos



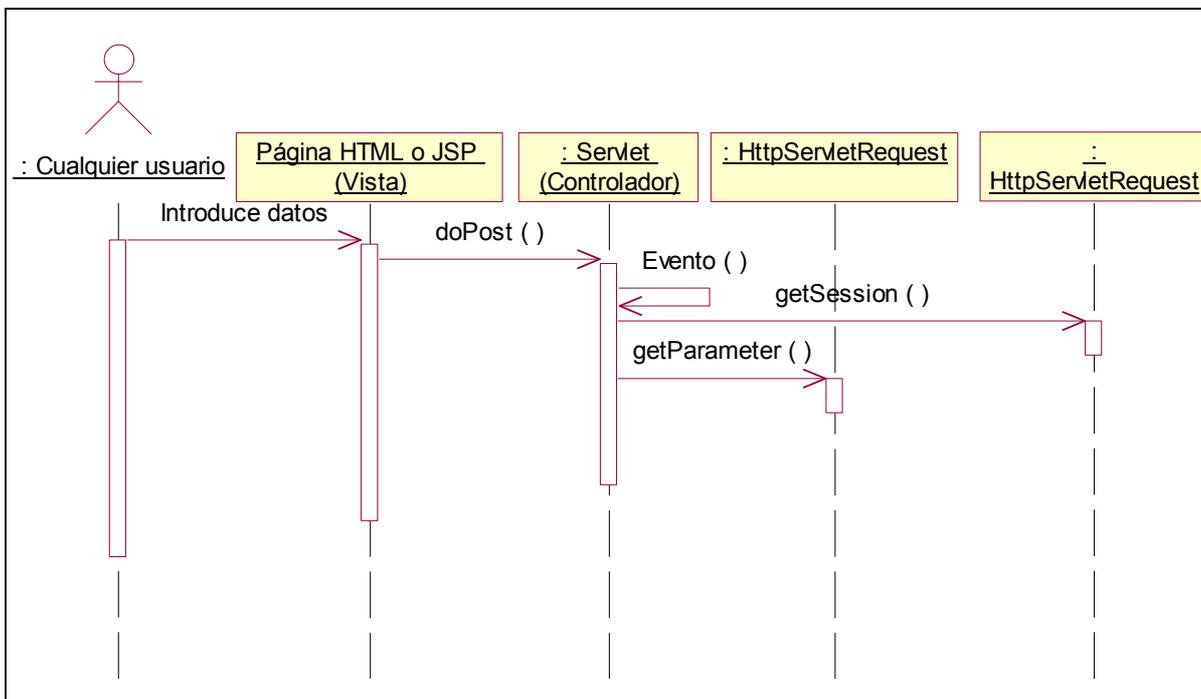
X) Desconexión de la base de datos de un bean con consultas almacenadas en páginas (PRS)



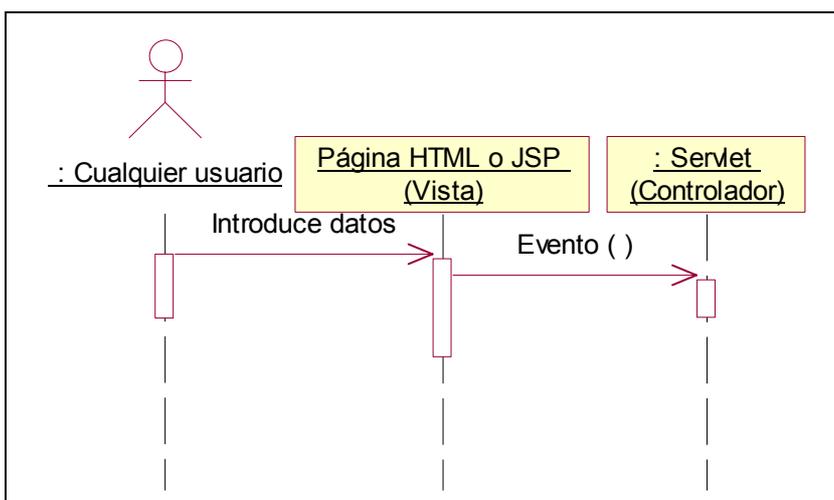
Controladores

También los abstraeremos, pues es un proceso repetitivo en todos los diagramas y así facilitaremos la comprensión del resto de diagramas de secuencia.

XI) Procesamiento de los eventos



Se mostrará como:

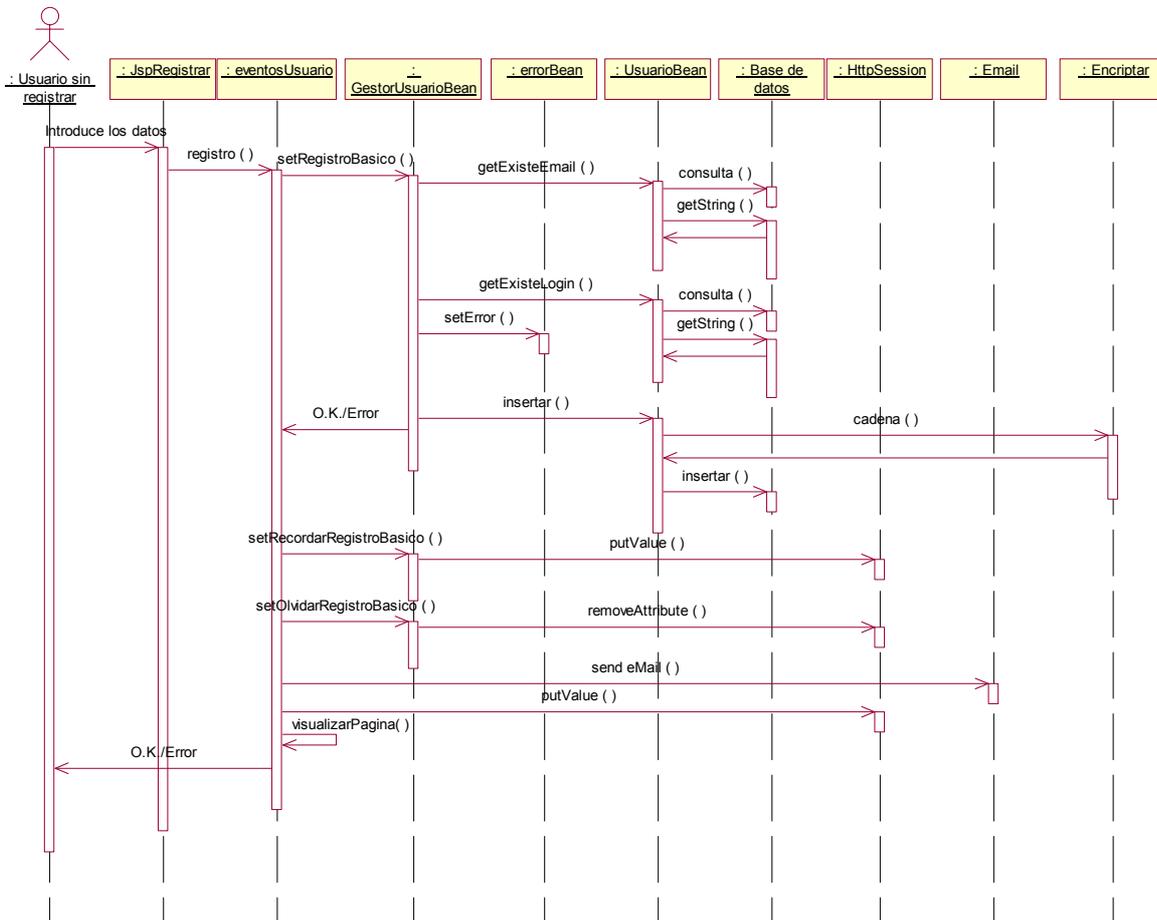


DIAGRAMAS DE SECUENCIA (correspondiente a los escenarios)

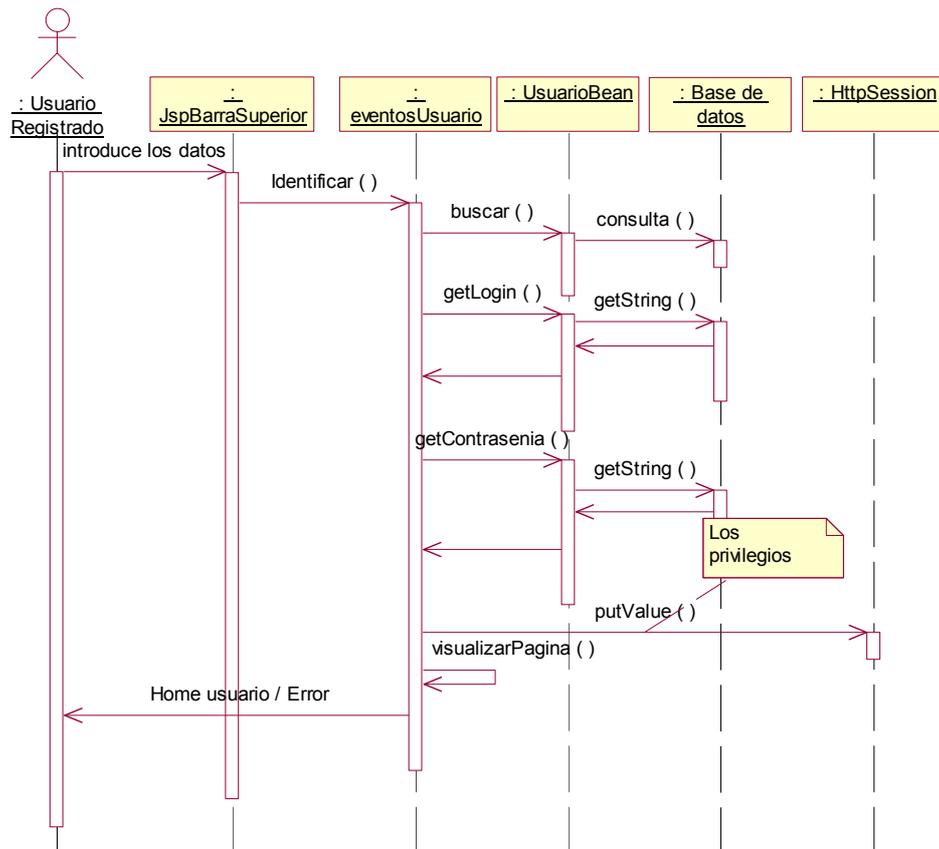
Para la comprensión de los siguientes diagramas habrá que recurrir a los descritos más arriba para tener una visión más detallada del sistema. Pues se abstraerá cierta información para poder lograr una mayor comprensión por parte del lector, ya que si se muestran con total detalle podría llegar a ser confuso, por la gran cantidad de clases reflejadas y mensajes enviados entre ellas.

1) VALIDACIÓN DE USUARIOS

1.1) Registro Usuarios



1.2) Validación de Usuarios



2) ADMINISTRACIÓN LOCAL

Este caso de uso, deberá ser llevado a cabo por la herramienta con la que se gestiona la base de datos, si se trata de una base de datos Access, pues será este programa con el mismo nombre, el que nos permitirá administrar localmente la base de datos, por tanto no describiré los diagramas pues no formarán parte de la herramienta que estoy diseñando.

Se hizo el estudio en la etapa de análisis fue para proporcionar una noción global sobre las funciones que debería de desempeñar el administrador local de la herramienta. En esta etapa de diseño identificamos estos casos de uso como disponibles en las herramientas que nos dan soporte a la base de datos y por tanto siguiendo el *lema* “no reinventar la rueda” no diseñaremos ni implementaremos una herramienta de administración local de la base de datos.

3) ADMINISTRACIÓN REMOTA

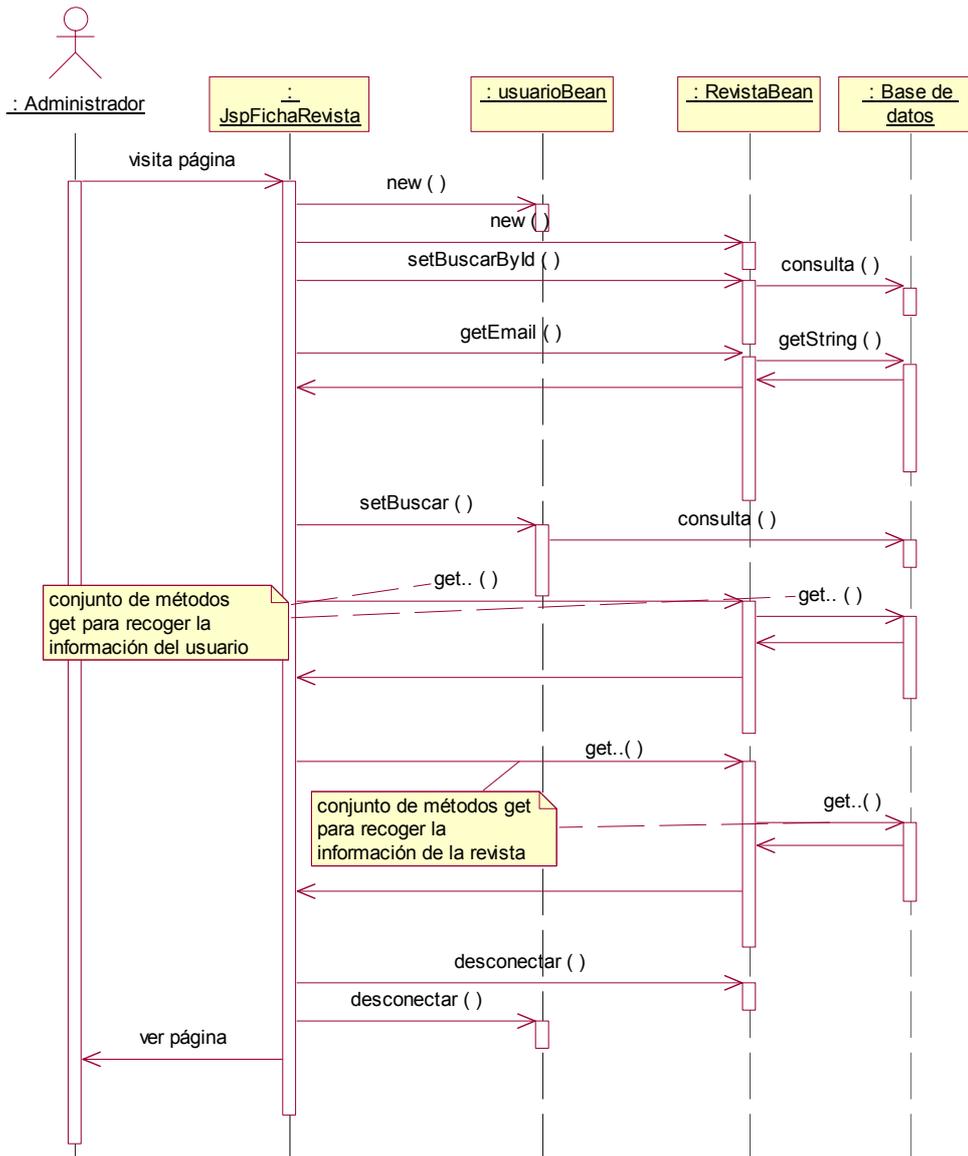
Ya se ha comentado en la fase de análisis que era un caso trivial, pues la herramienta que diseñamos por si misma ya realiza la administración remota y por tanto el estudio de esta se hace totalmente innecesario.

4) CREAR, MODIFICAR Y ELIMINAR REVISTAS

4.1) Ver información sobre una revista

Descompondré este diagrama en dos, uno para cada tipo de usuario, ya que se mostrará la información de manera distinta.

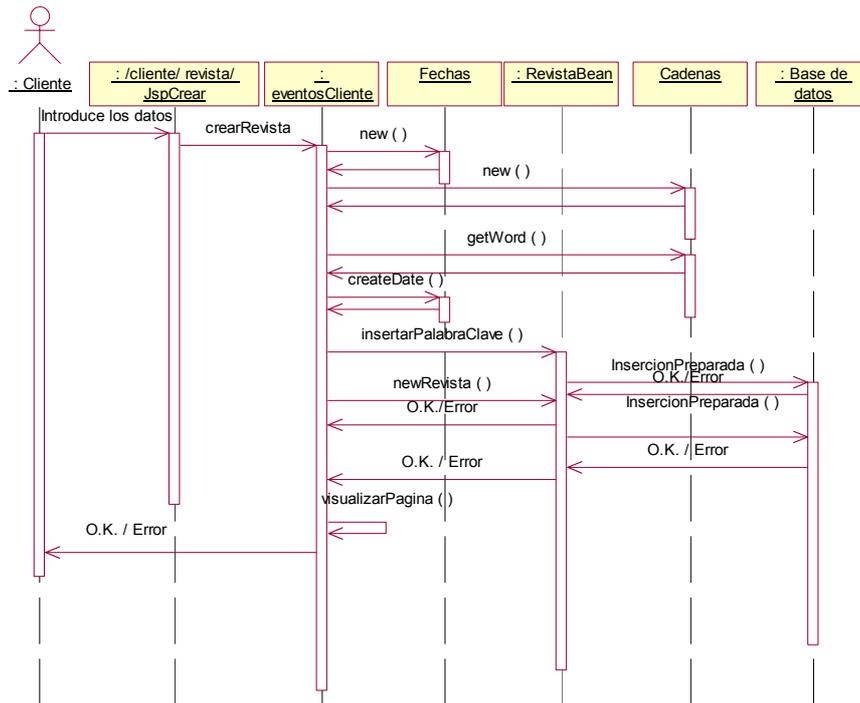
4.1.1. Por un administrador



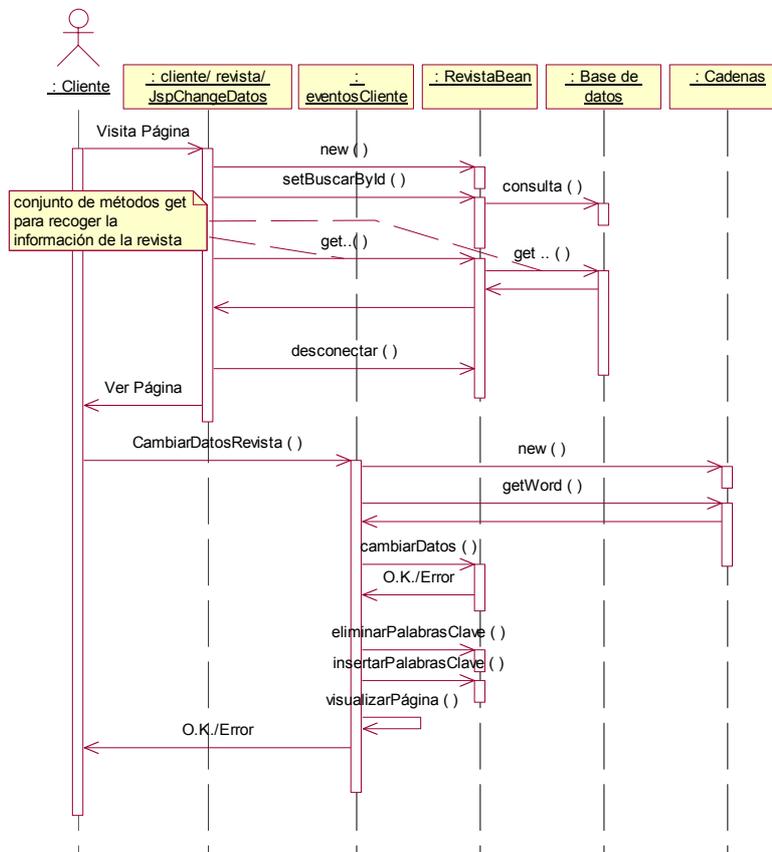
4.1.2 Por un Cliente

El diagrama sería igual, pero iniciaría el proceso el Cliente y `: JspFichaRevista` sería `:/cliente/revista/JspFicha`

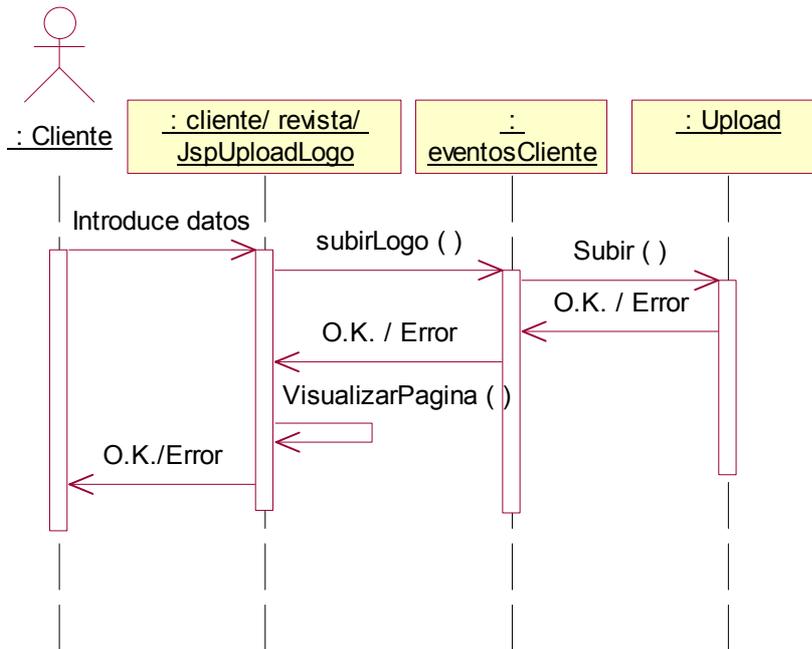
4.2 Crear una Revista



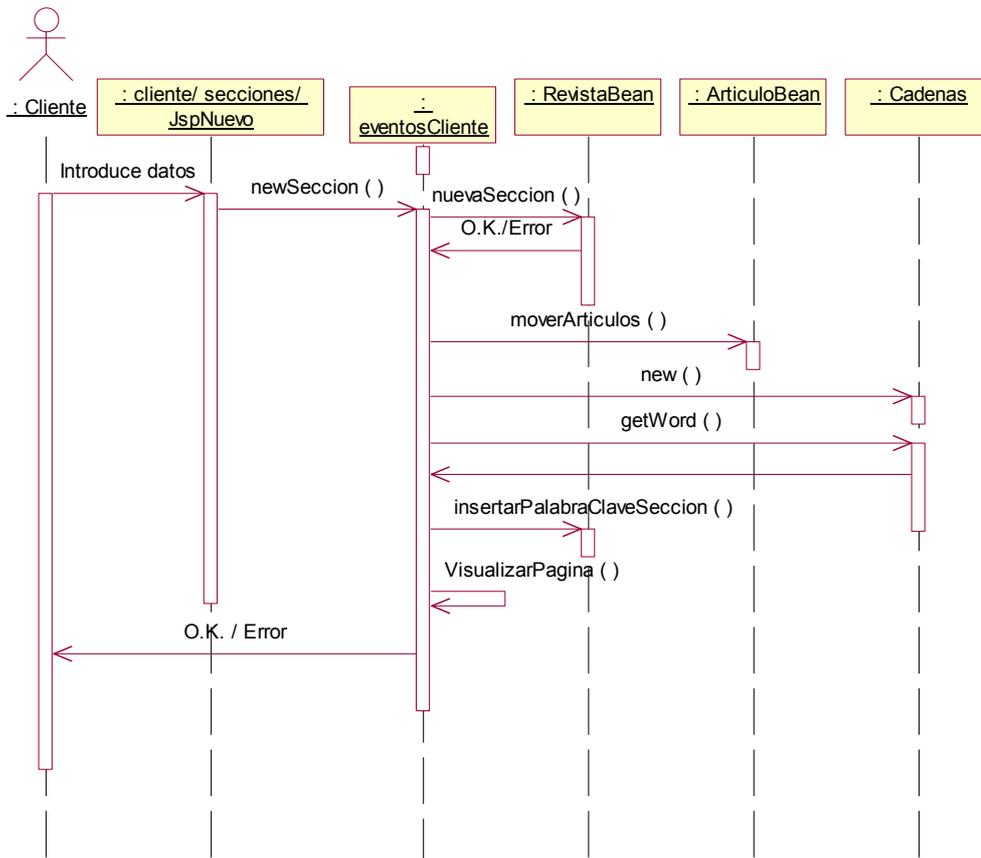
4.3 Modificar una revista



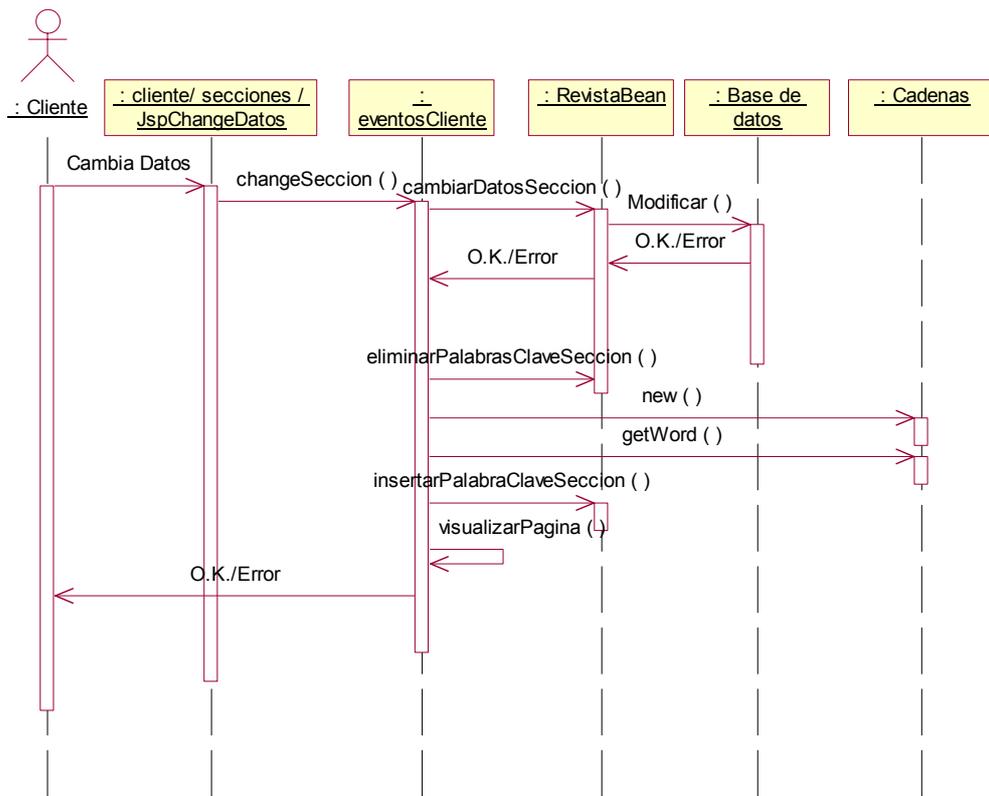
4.4 Subir Logotipo



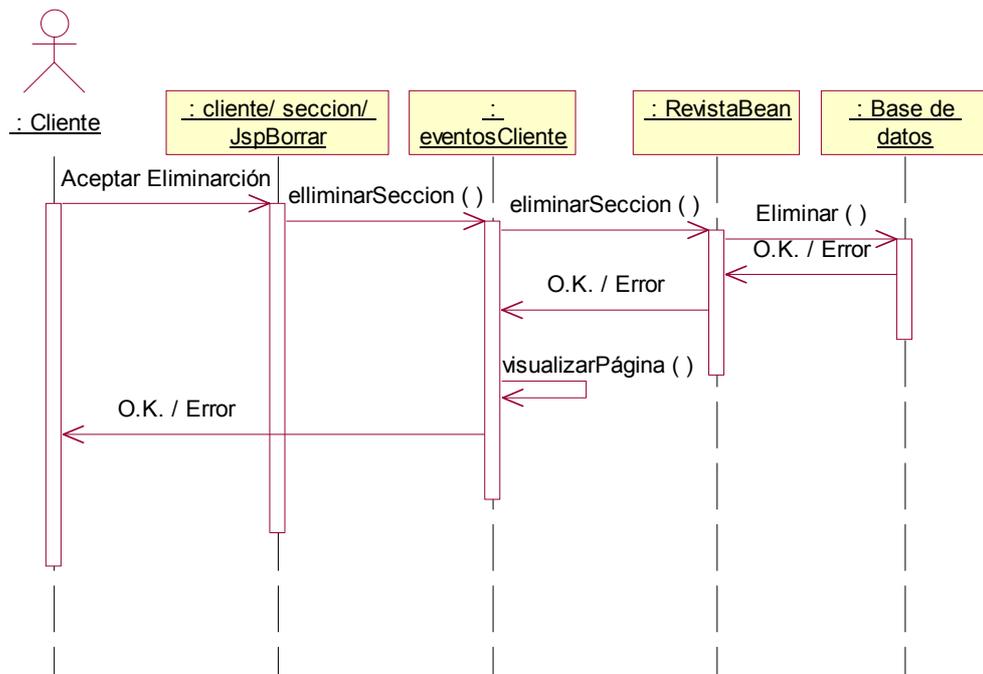
4.5 Crear Sección



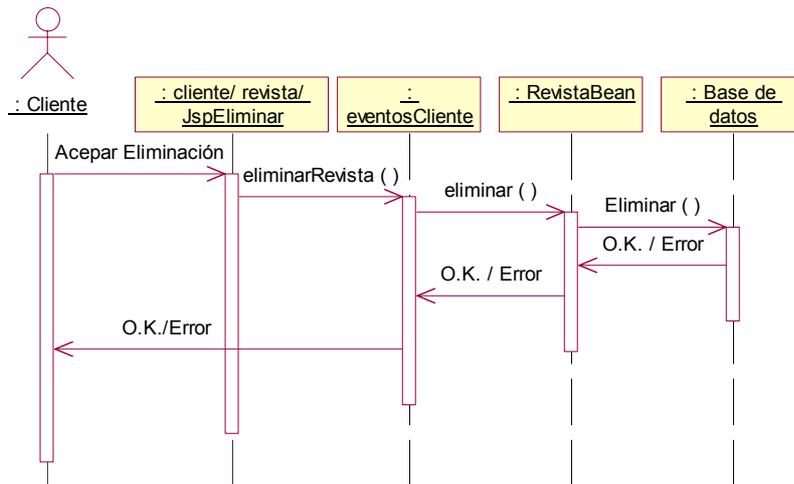
4.6 Modificar sección



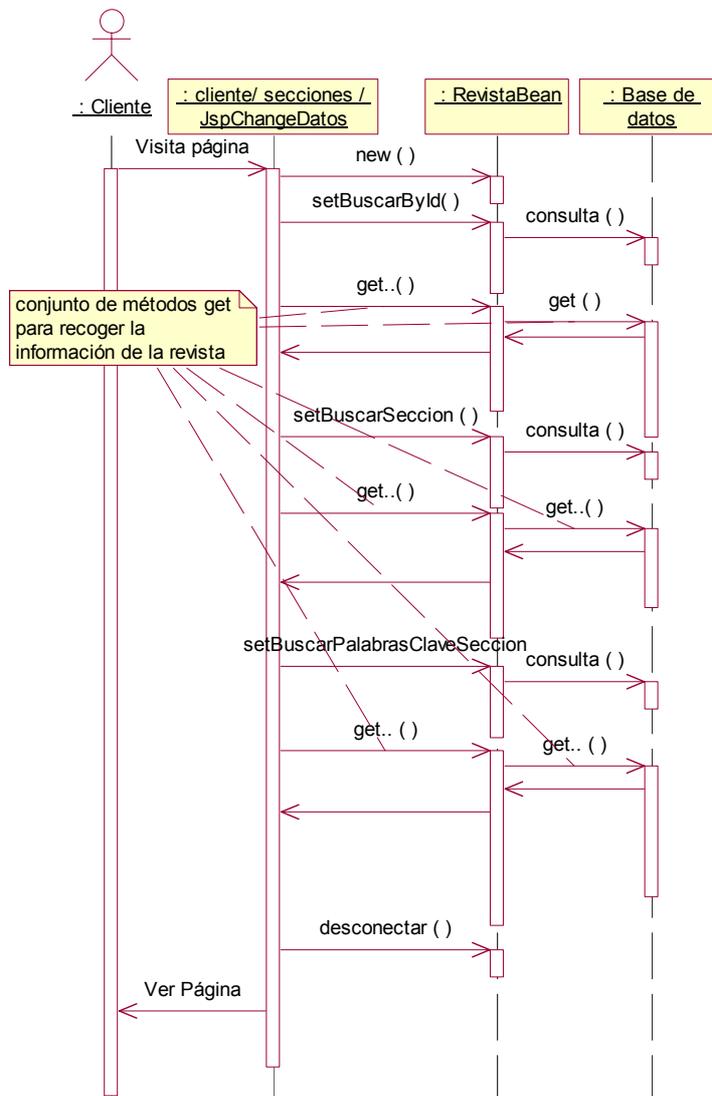
4.7 Eliminar Sección



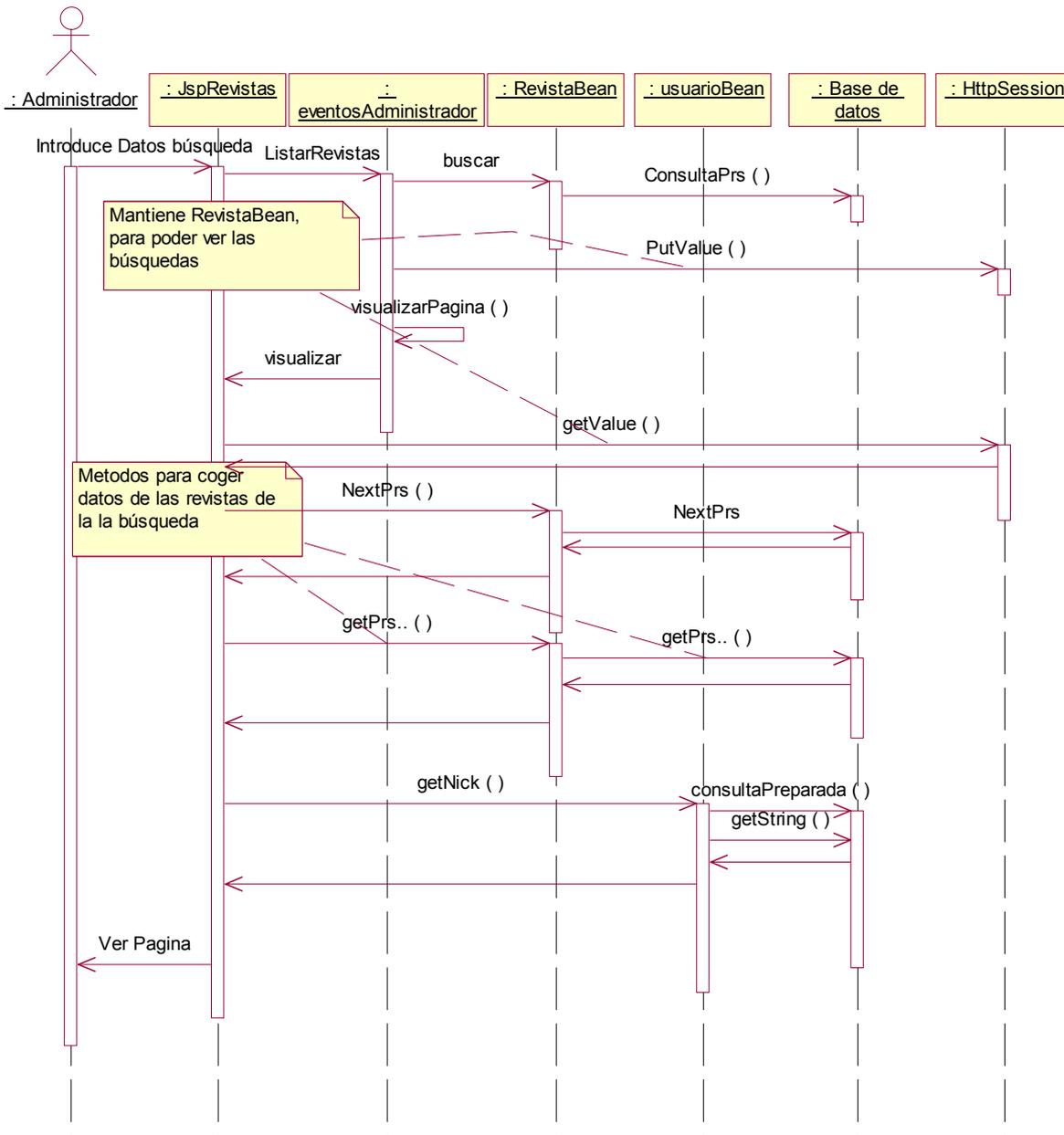
4.8. Eliminar revista



4.9 Ver información de una sección

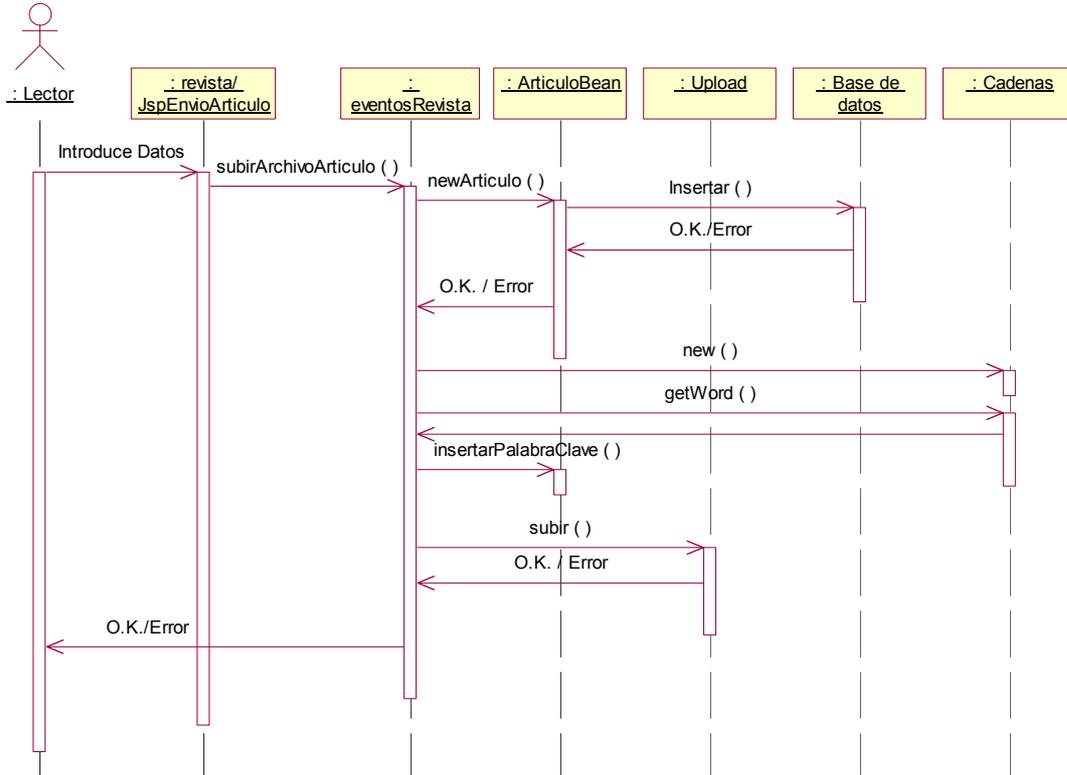


4.10 Listar revistas

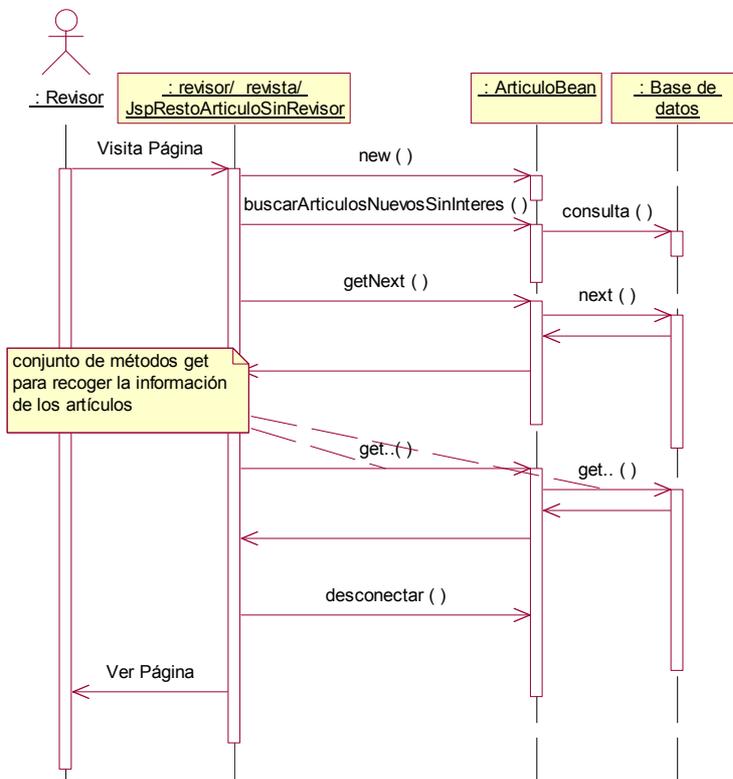


5) GESTIÓN ARTÍCULOS

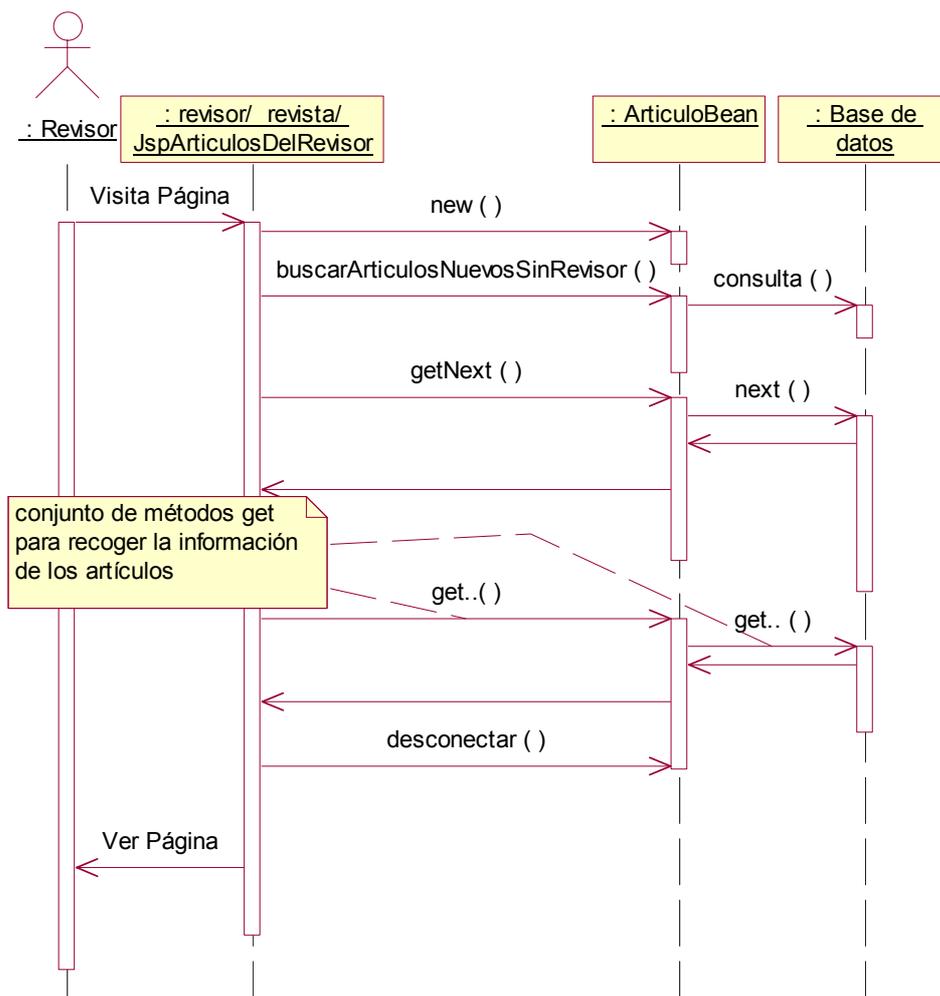
5.1) Enviar artículo



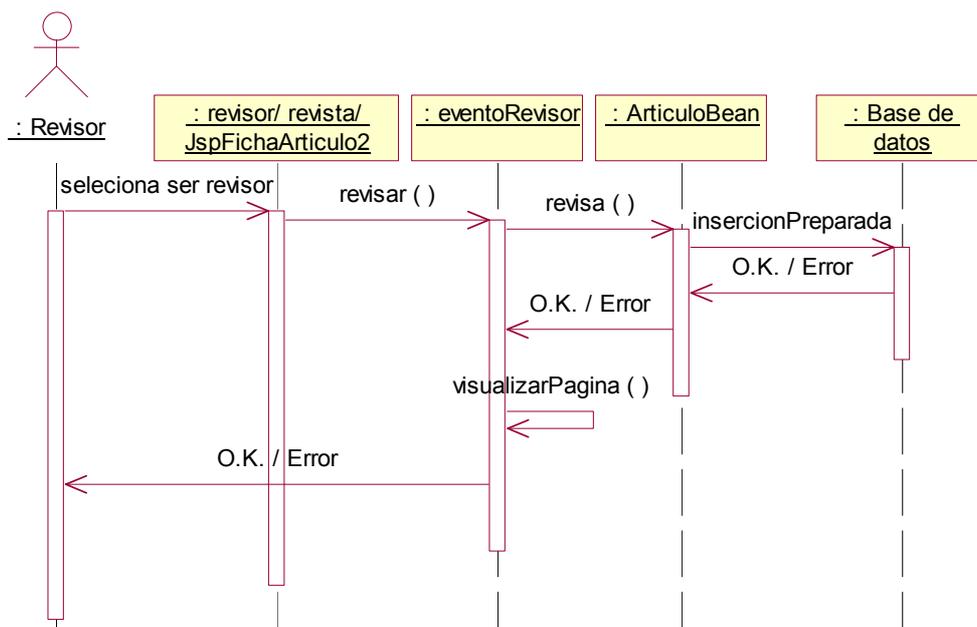
5.2 Ver artículos sin revisor



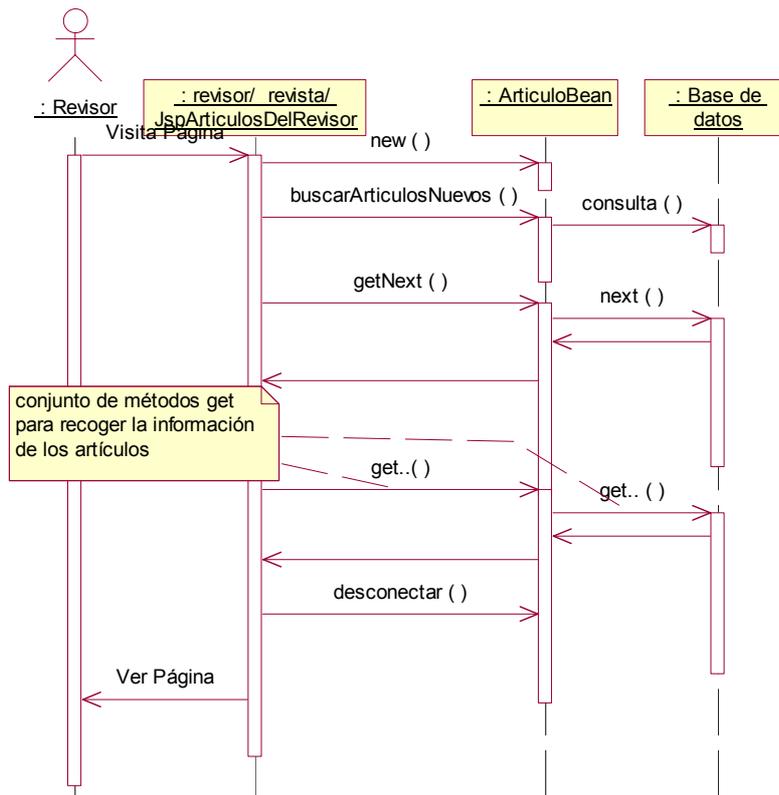
5.3 Ver artículos sin revisor que interesan



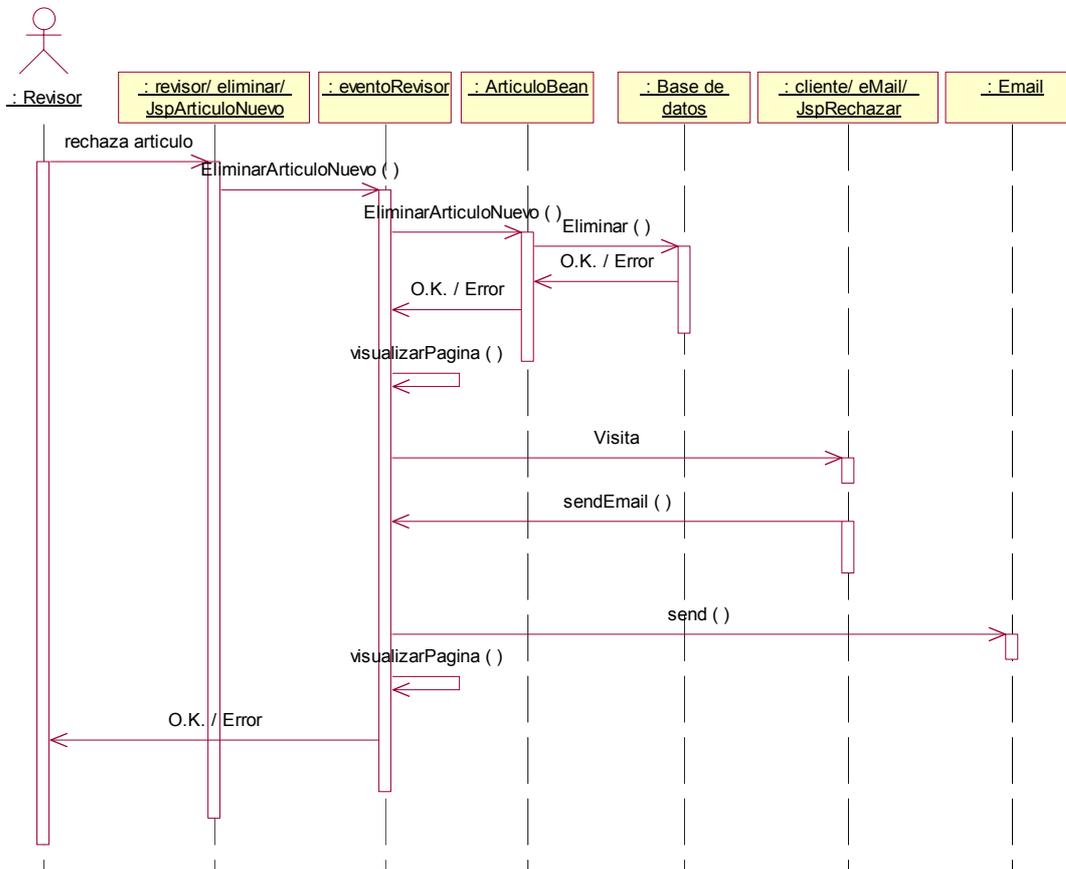
5.4 Asociar revisor con artículo



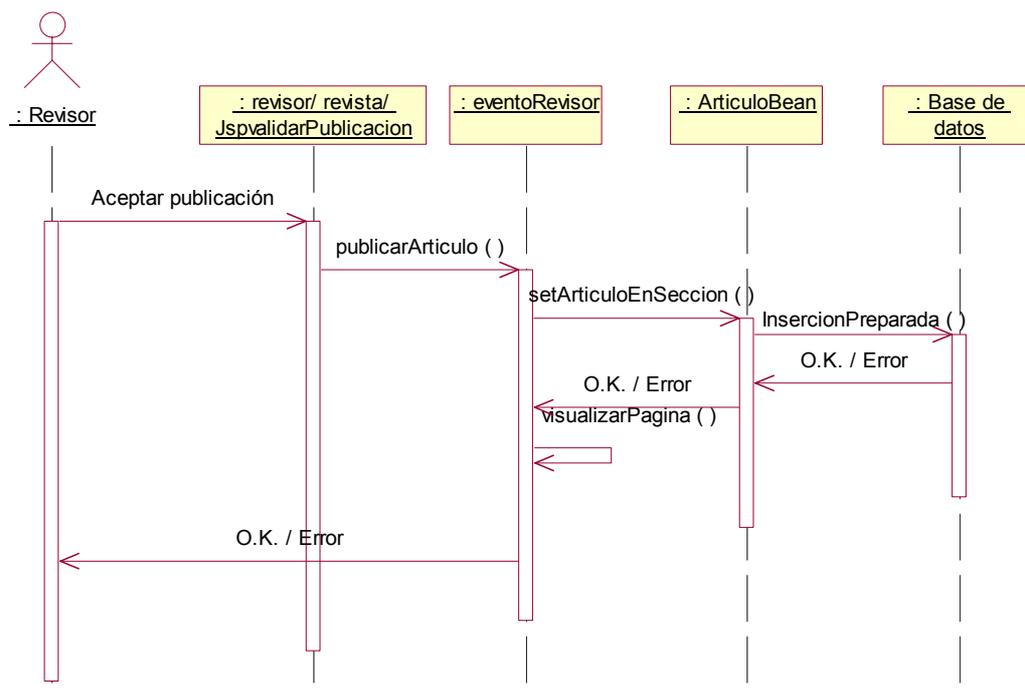
5.5 Ver artículos que revisas



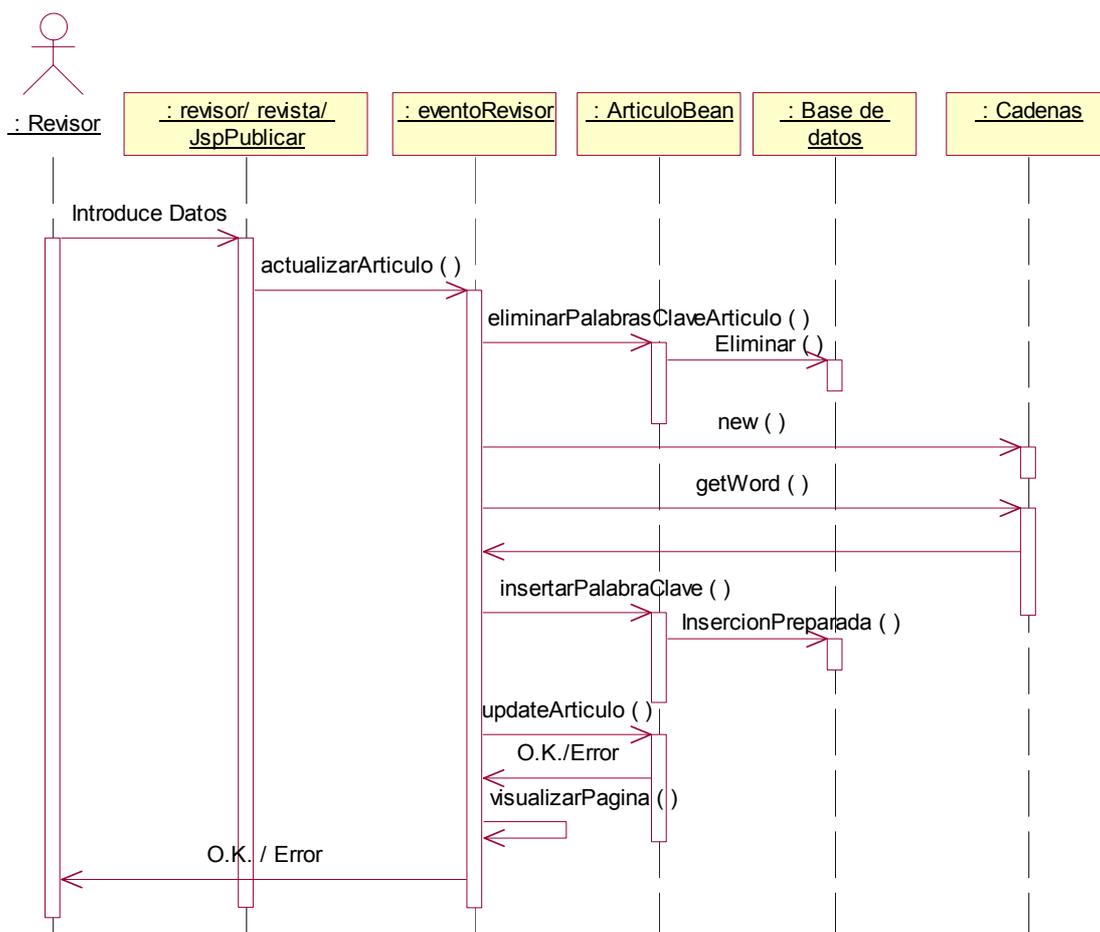
5.6 Rechazar artículo



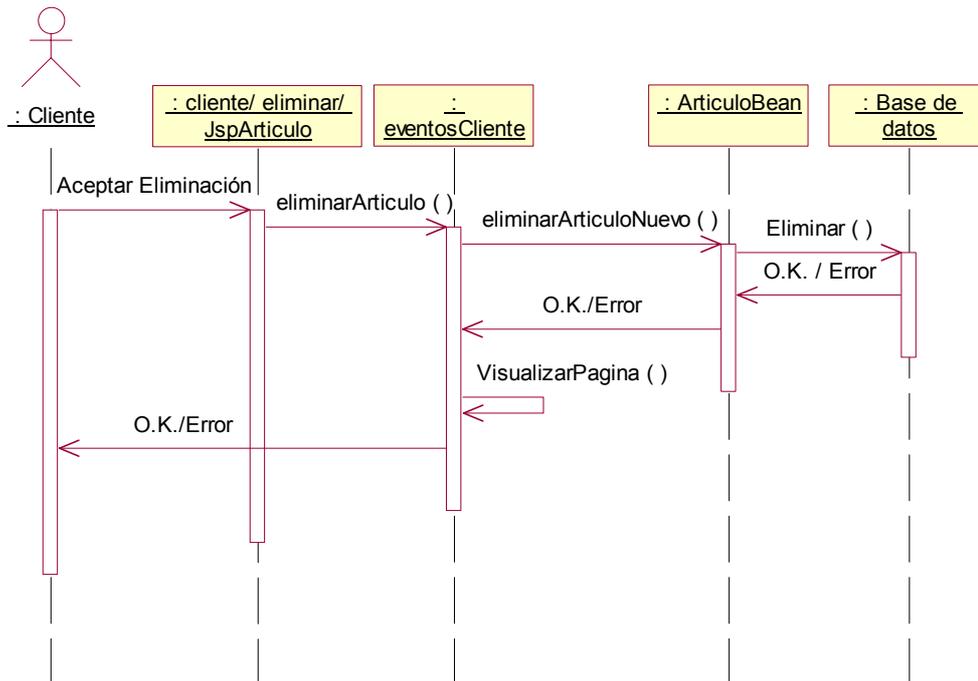
5.7. Publicar artículo



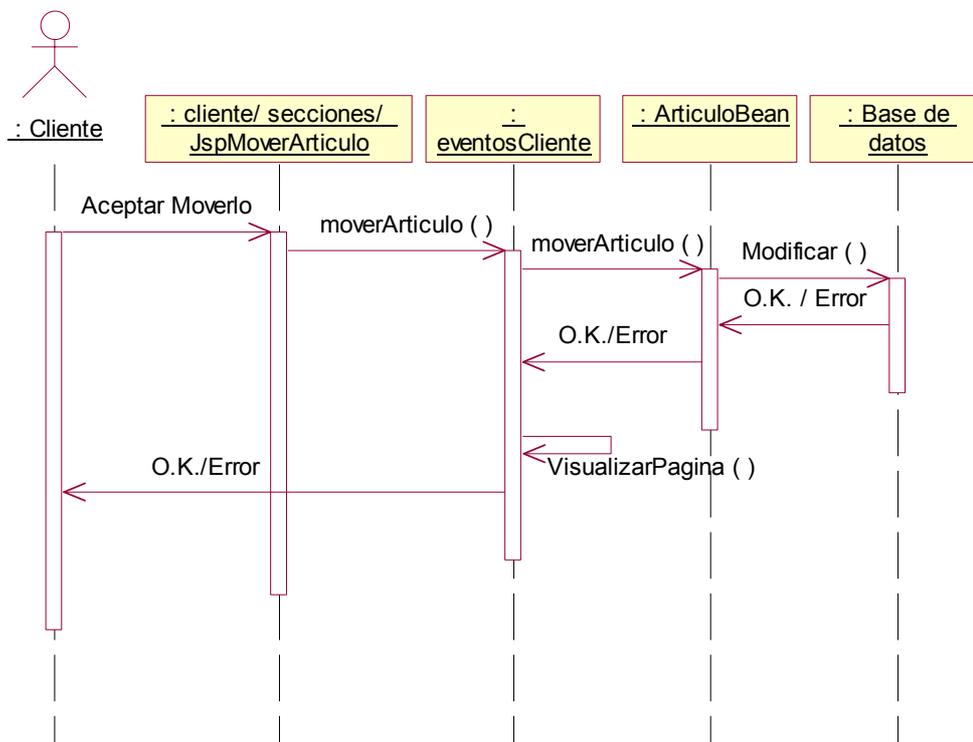
5.8 Cambiar datos del artículo



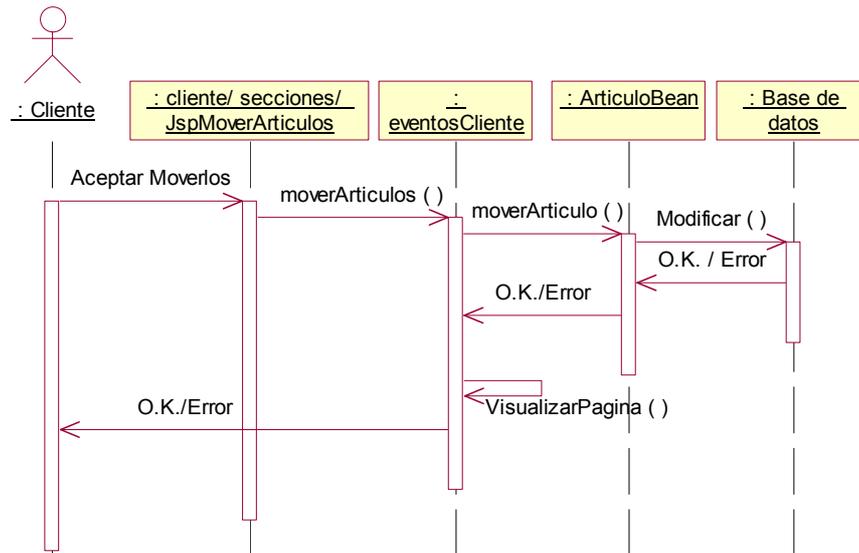
5.9 Eliminar artículo



5.10 Mover Artículo



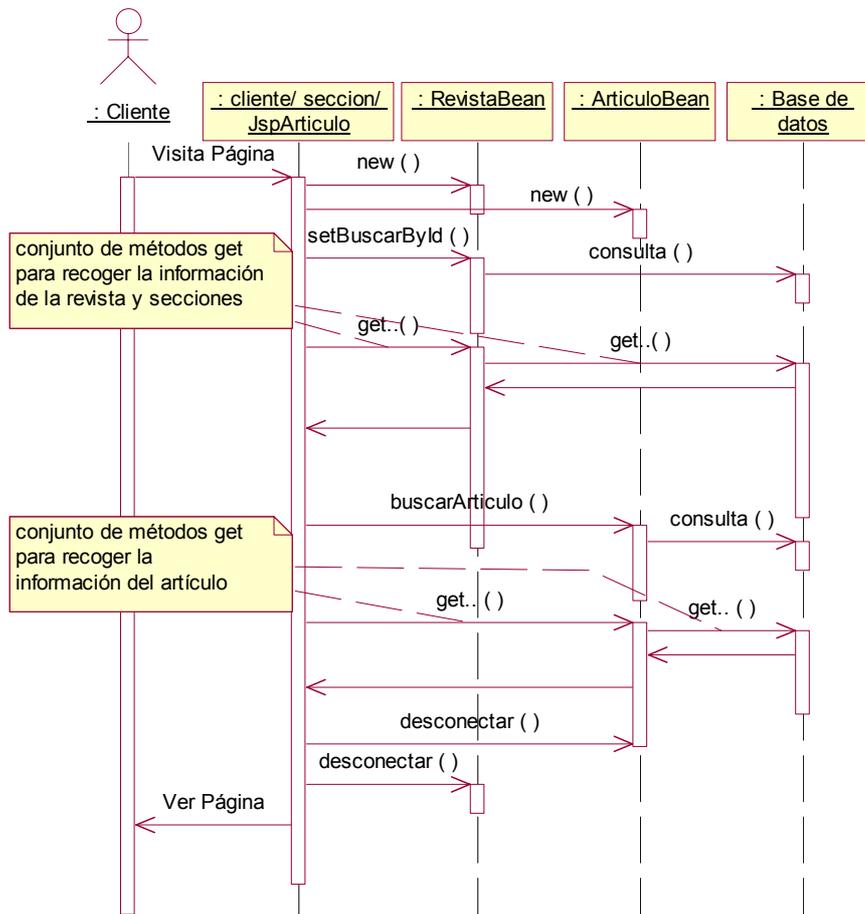
5.11 Mover todos los artículos de una sección



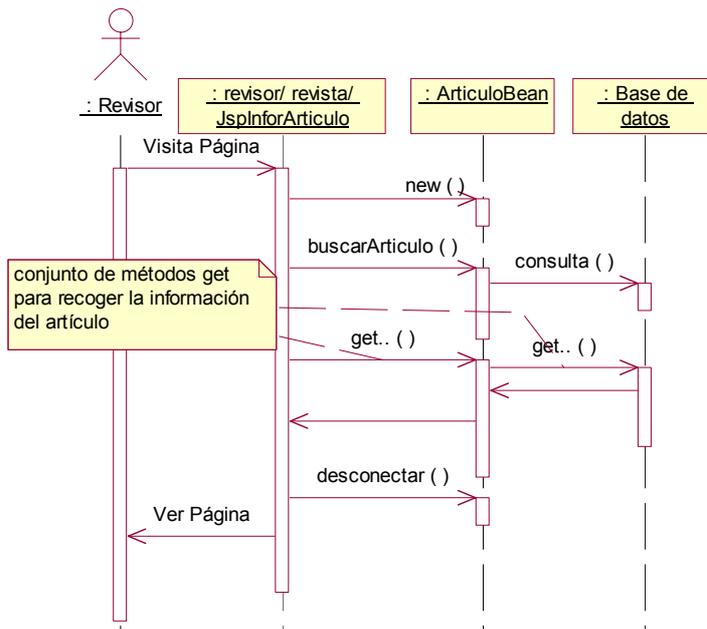
5.12 Ver datos del artículo

Aquí se dividirá en dos diagramas, uno por cada usuario que lo puede iniciar. Ya que cada usuario podrá ver los datos del artículo desde un contexto distinto.

5.12.1 Iniciado por el Cliente



5.12.2 Iniciado por el Revisor

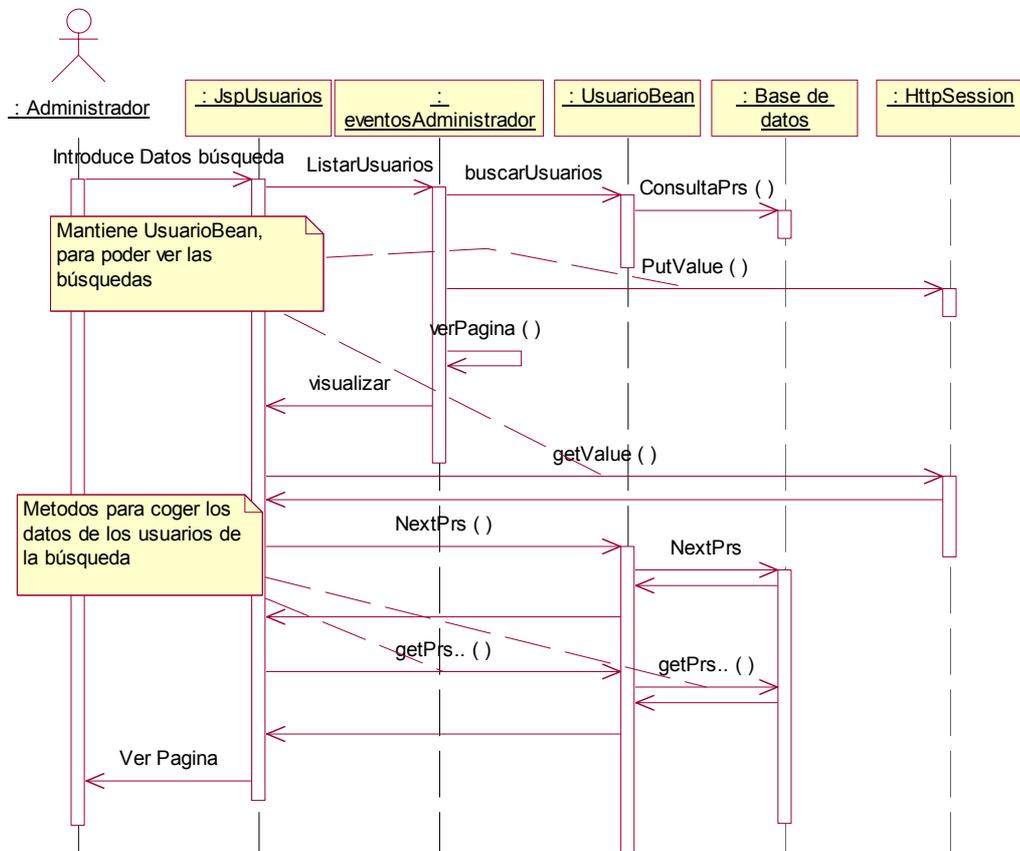


6) Gestionar Usuarios

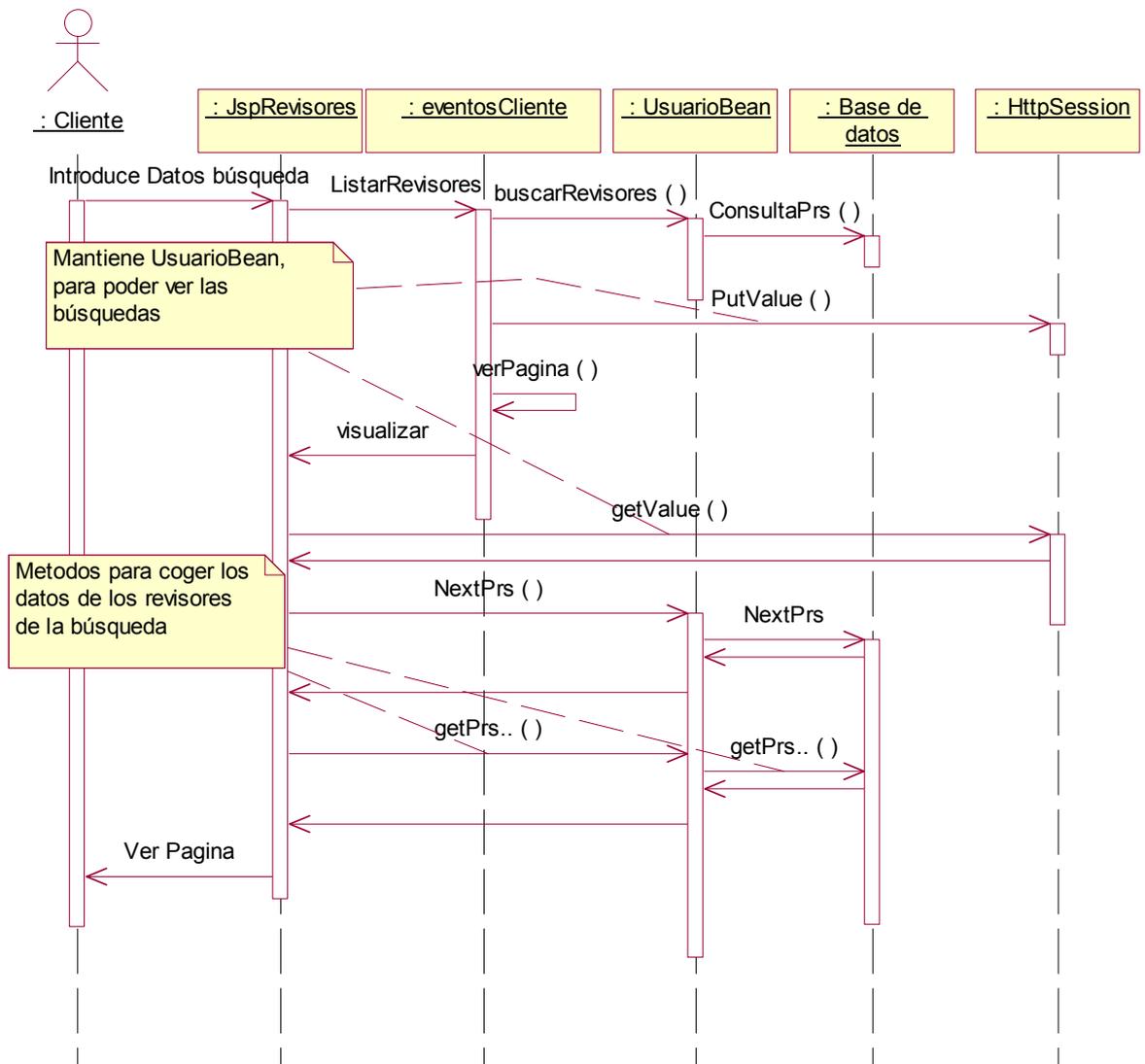
6.1 Listado de usuarios

Dividiremos entre los dos usuarios que lo pueden iniciar.

6.1.1. Iniciado por Administrador



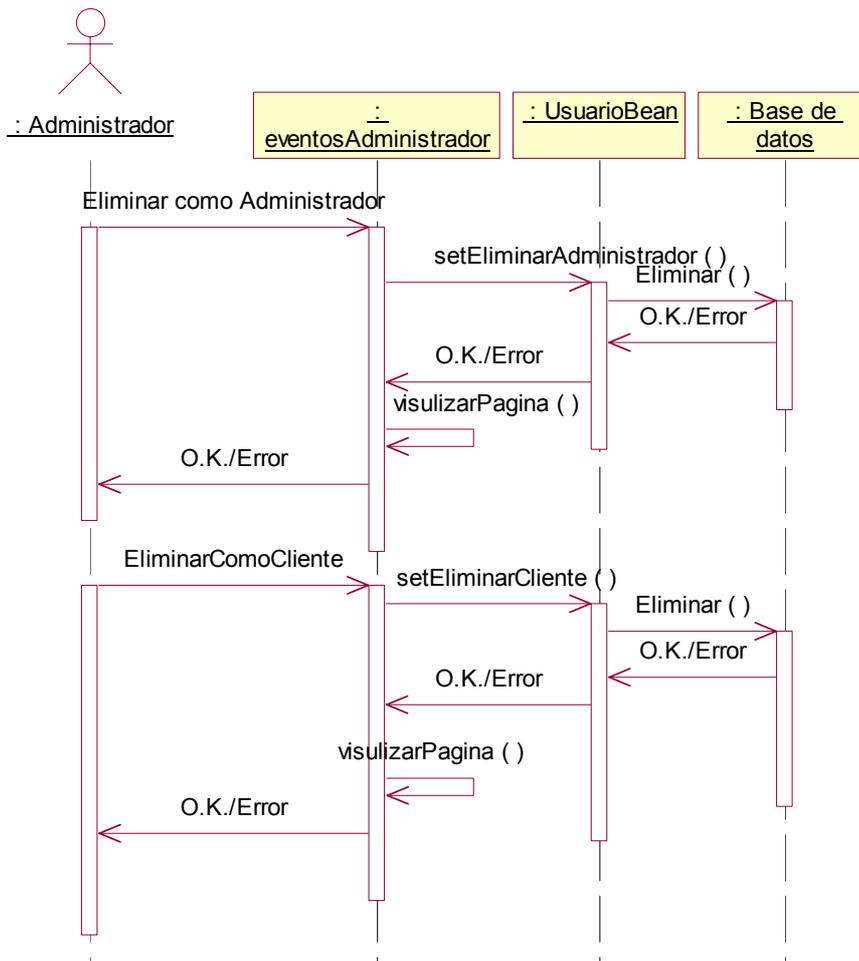
6.1.2. Iniciado por Cliente



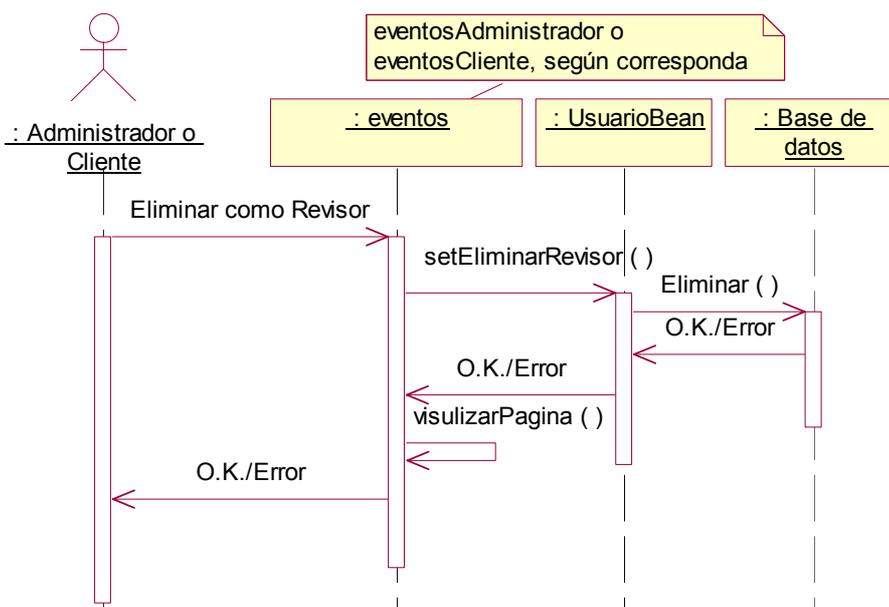
6.2 Quitar privilegio usuario

Como con el diagrama anterior, dividiremos este en dos, uno para eliminar un cliente o Administrador, que solo lo podrá realizar un Administrador.

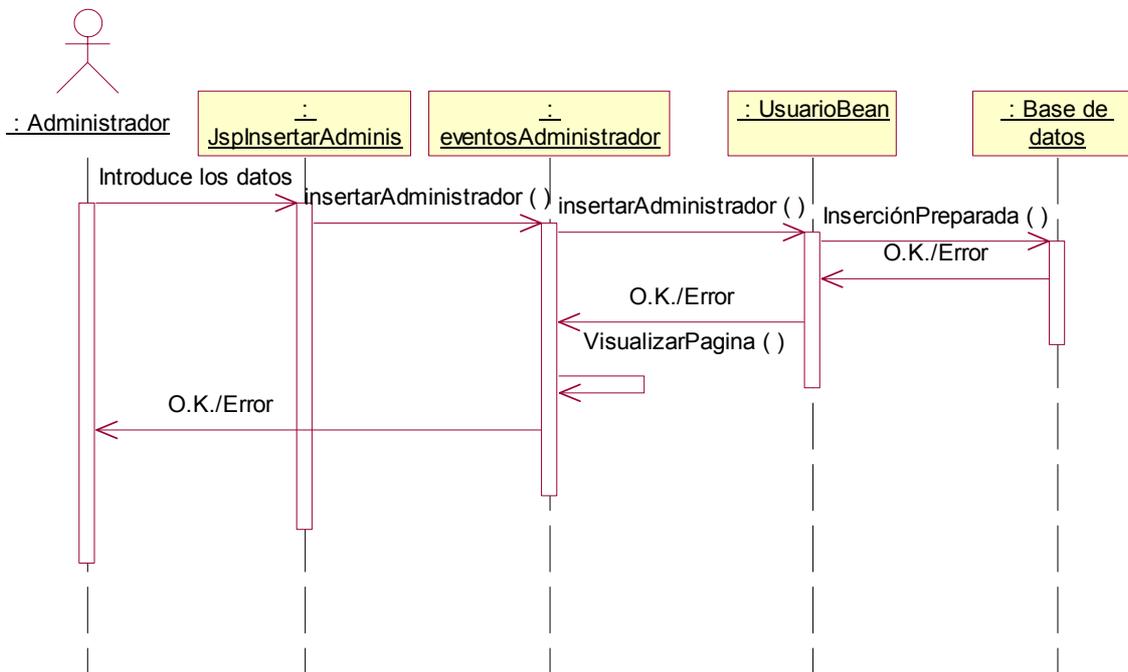
6.2.1. Eliminar como cliente o Administrador



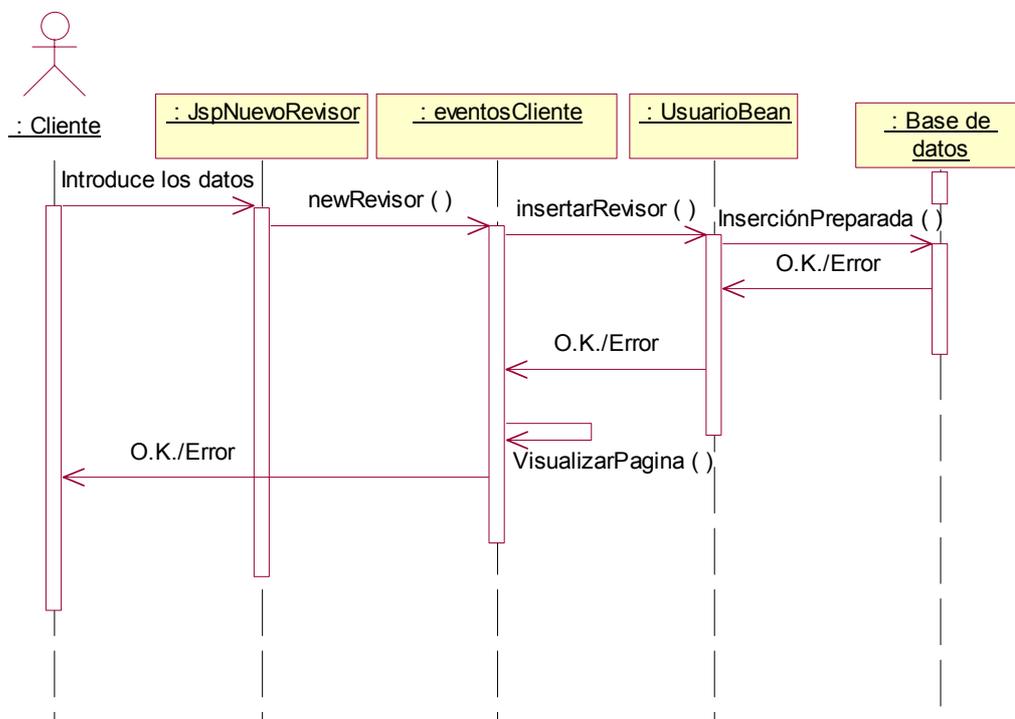
6.2.2. Eliminar como revisor



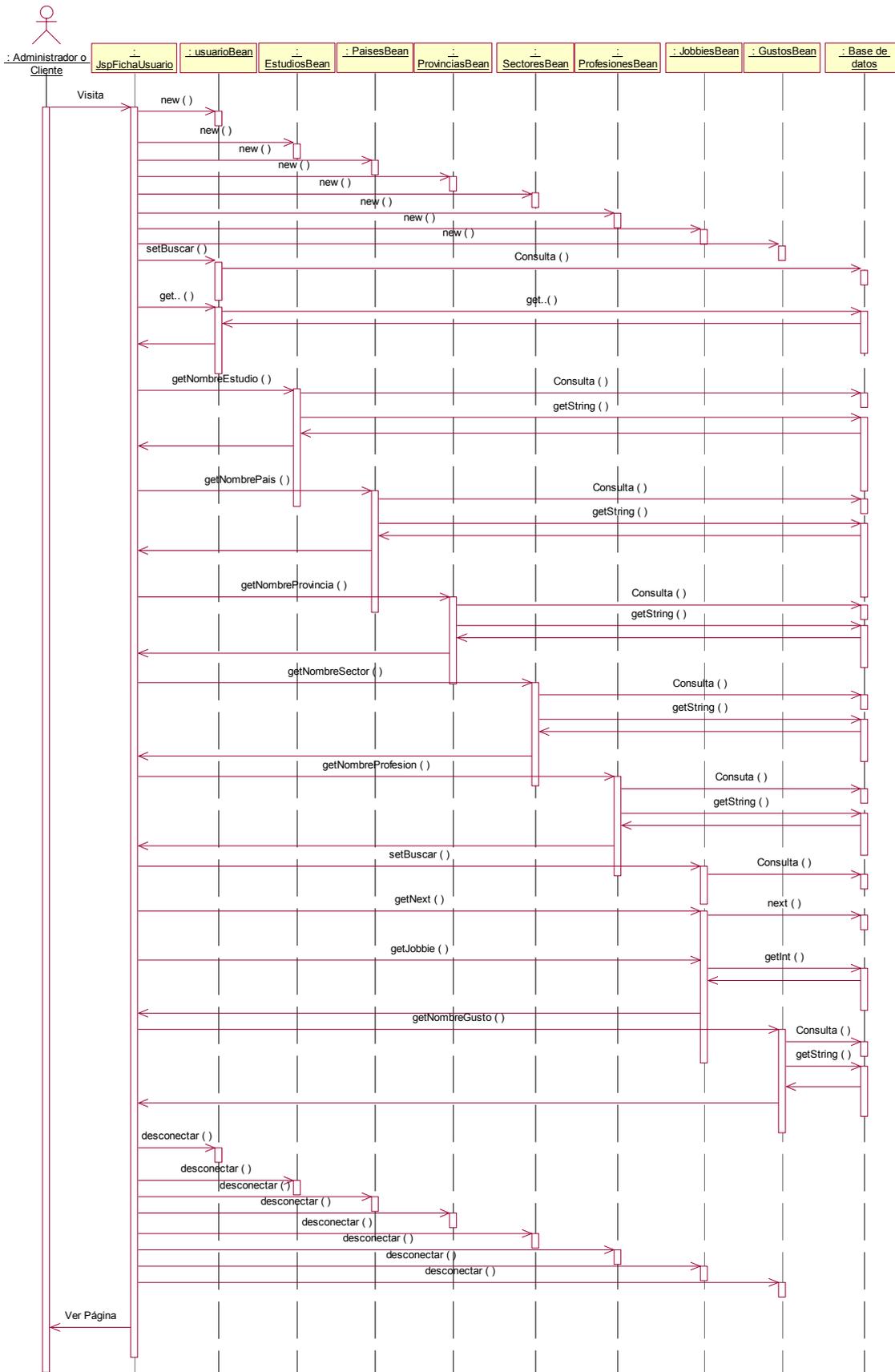
6.3 Insertar Administrador



6.4 Insertar Revisor

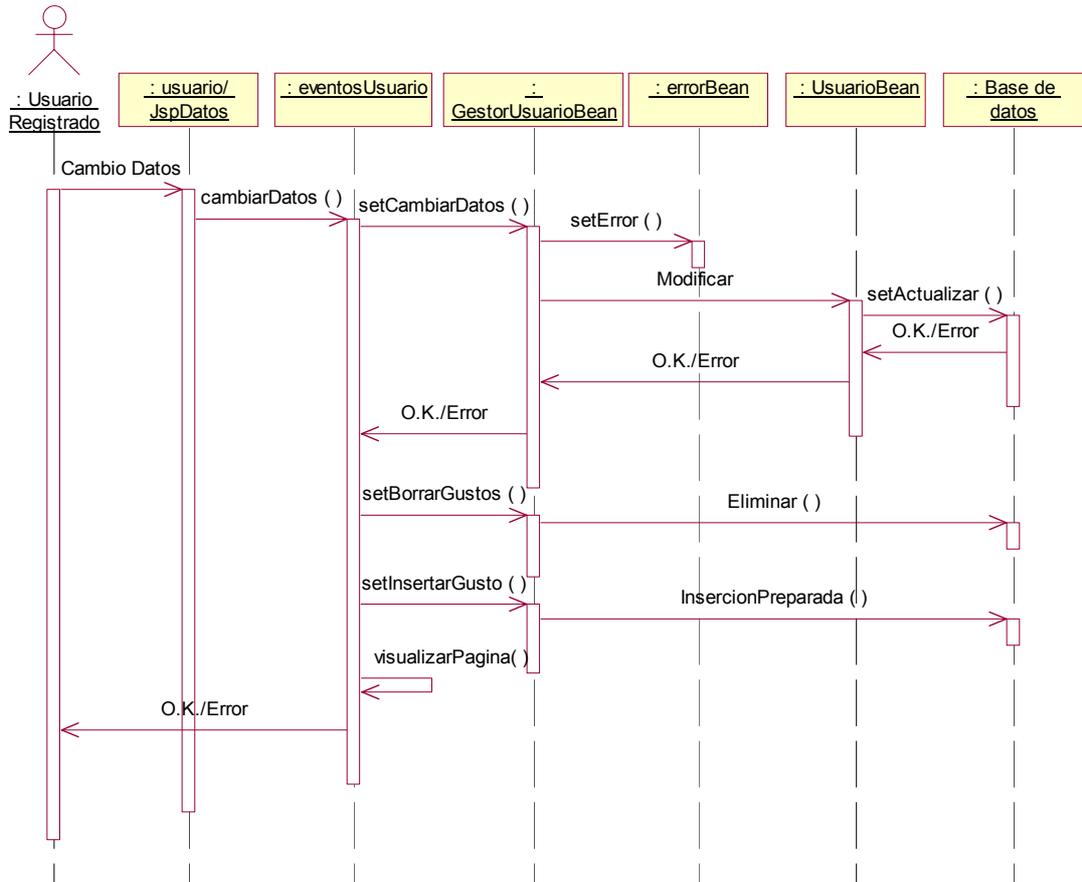


6.5 Ver información de un usuario

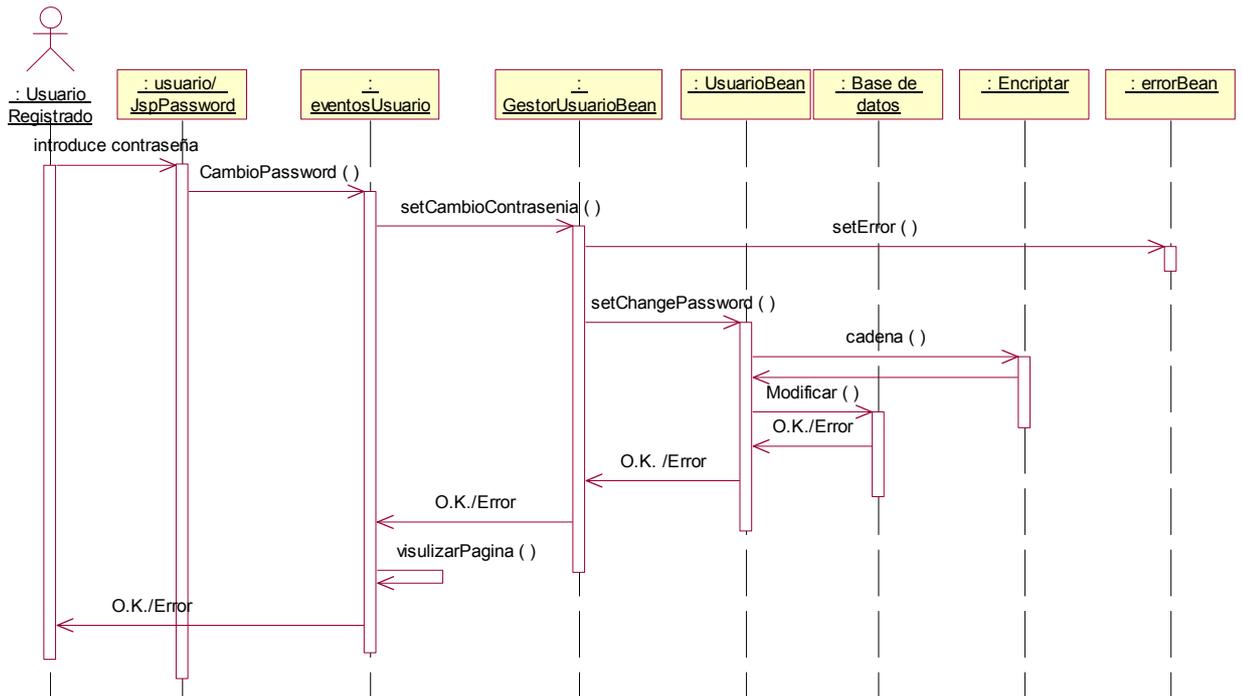


7) MODIFICAR PERFIL

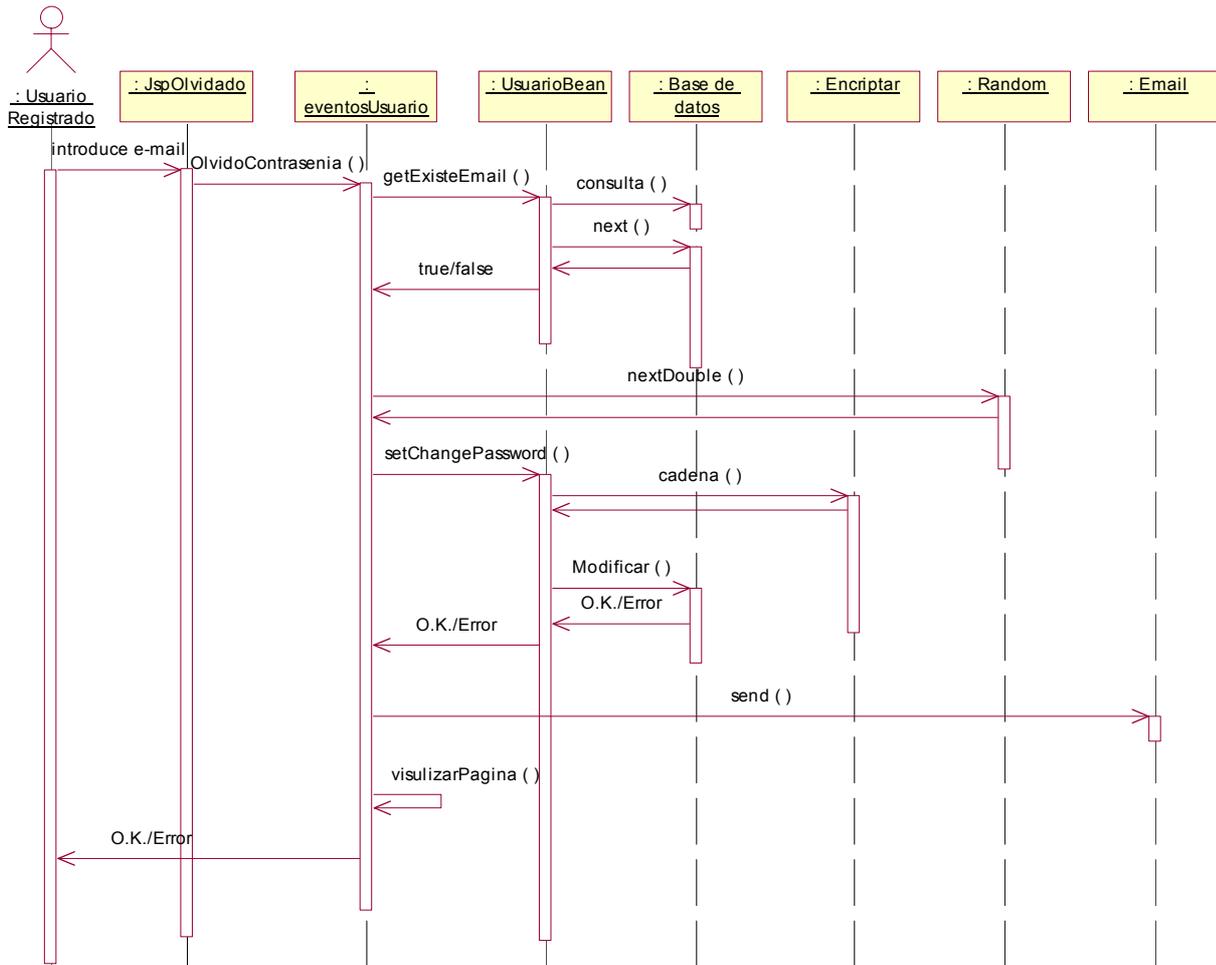
7.1 Modificar datos del usuario



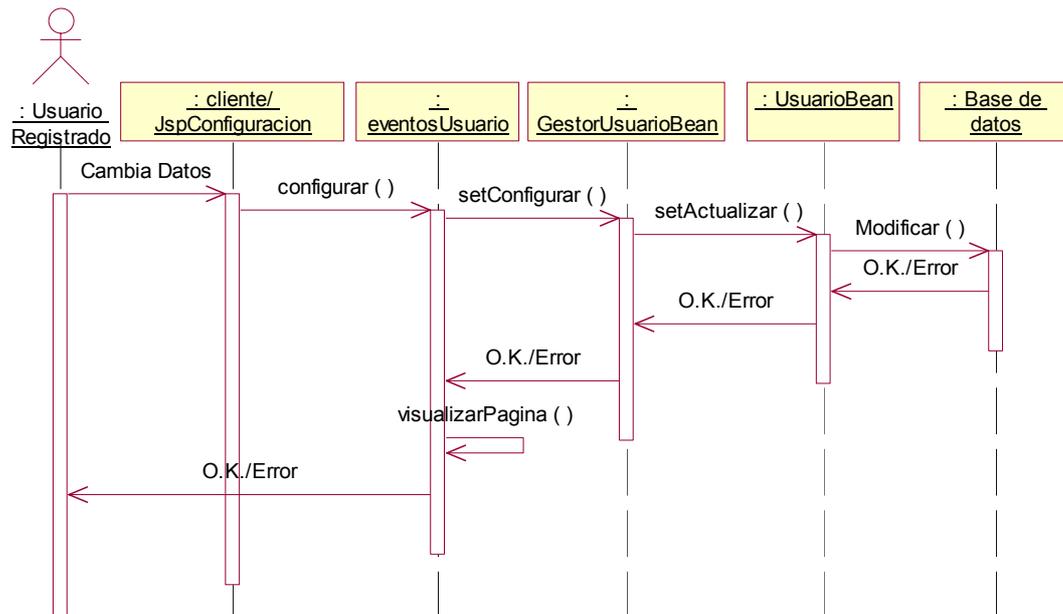
7.2 Cambiar Contraseña



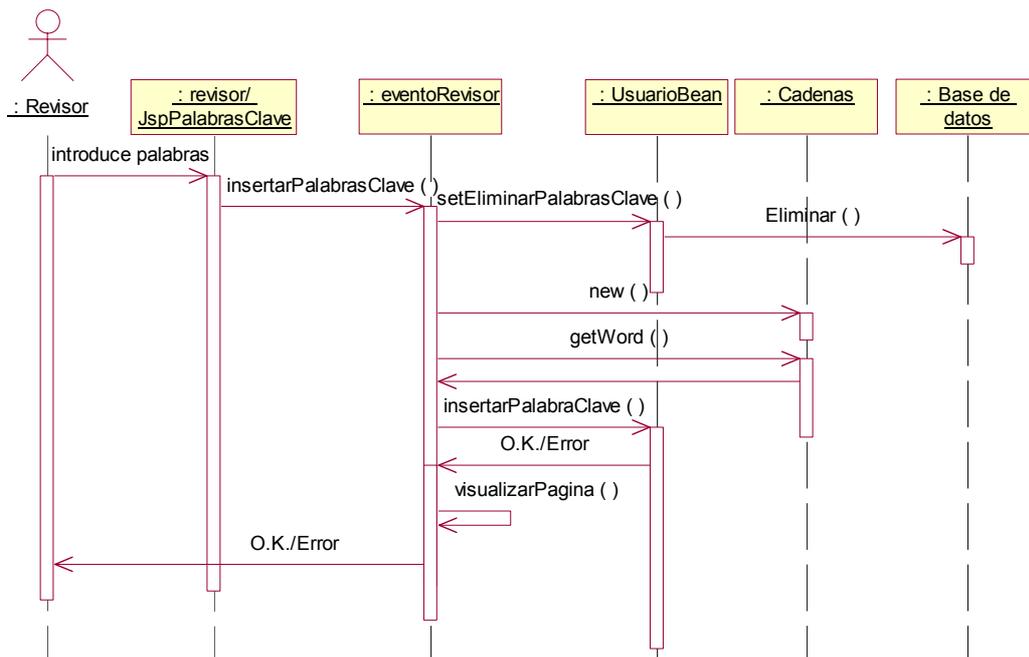
7.3 Enviar nueva contraseña



7.4 Configurar o personalizar

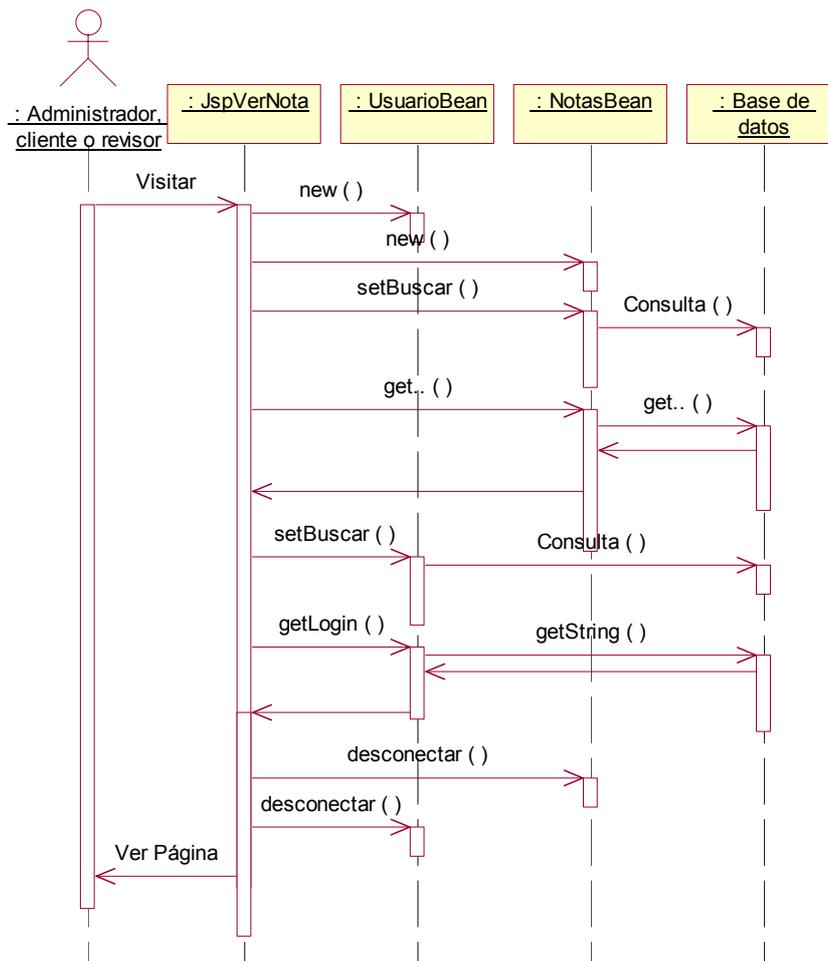


7.5 Asociar palabras claves

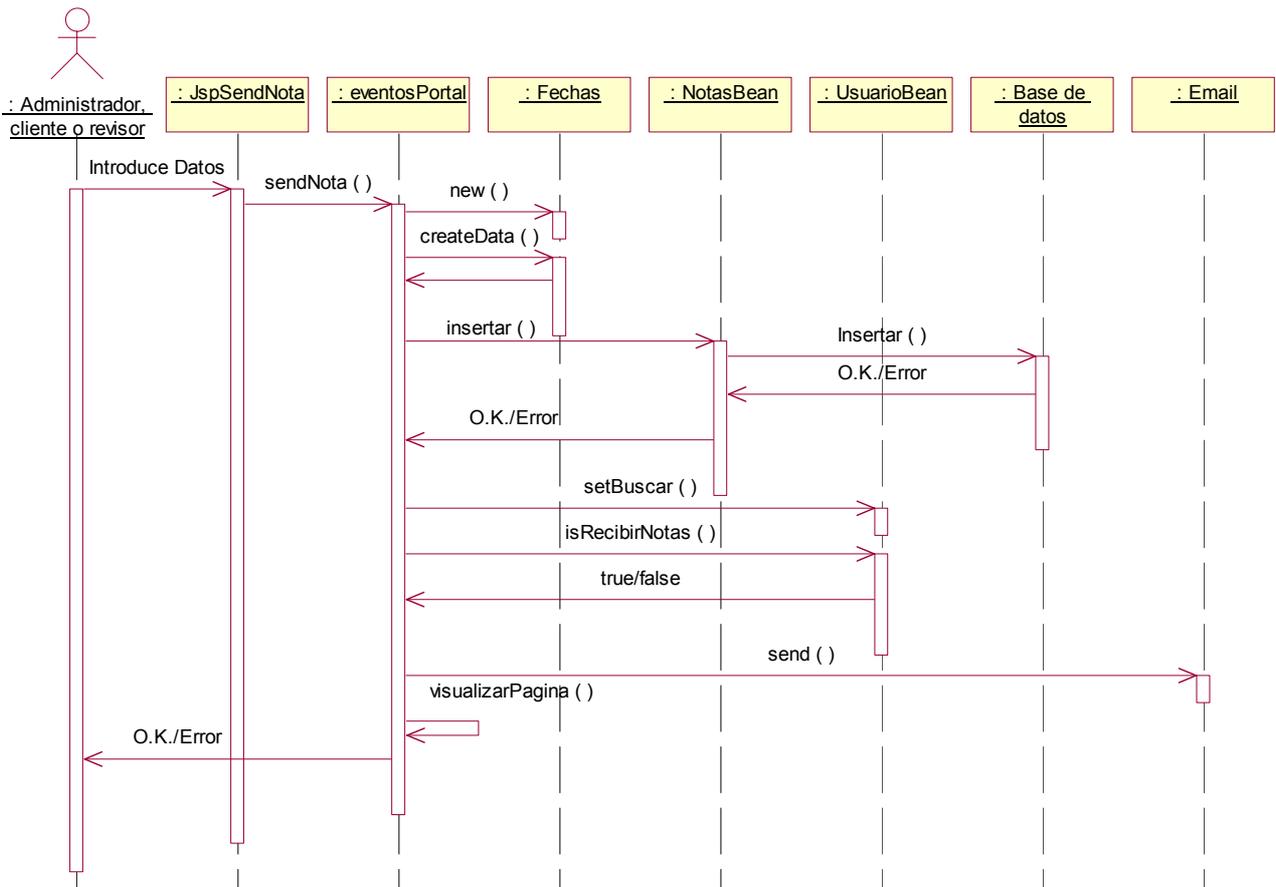


8) Comunicarse

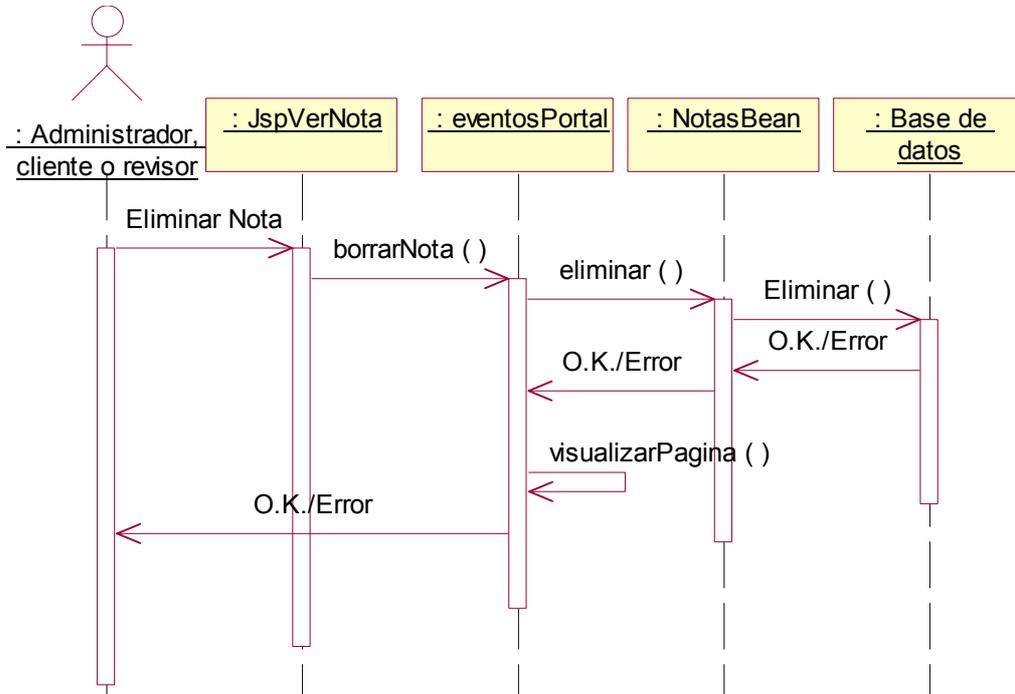
8.1 Leer Nota



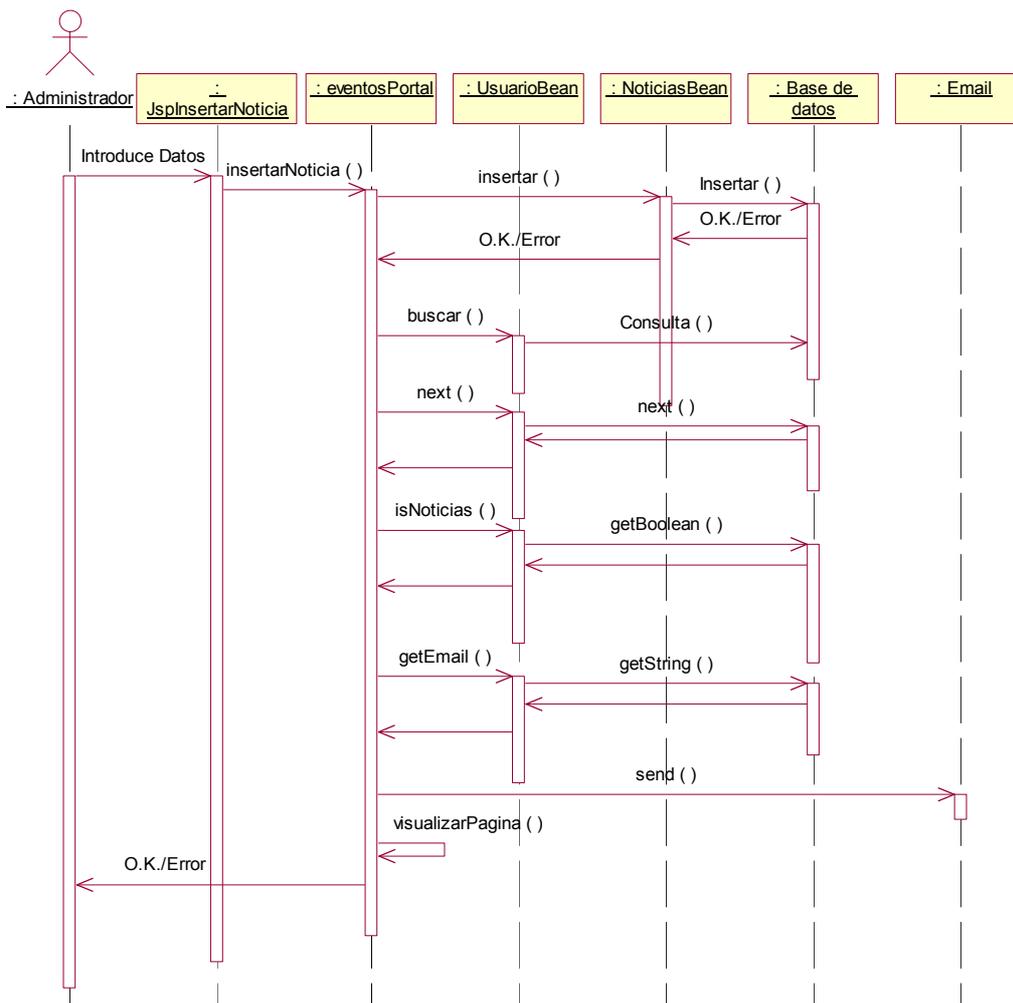
8.2 Enviar Nota



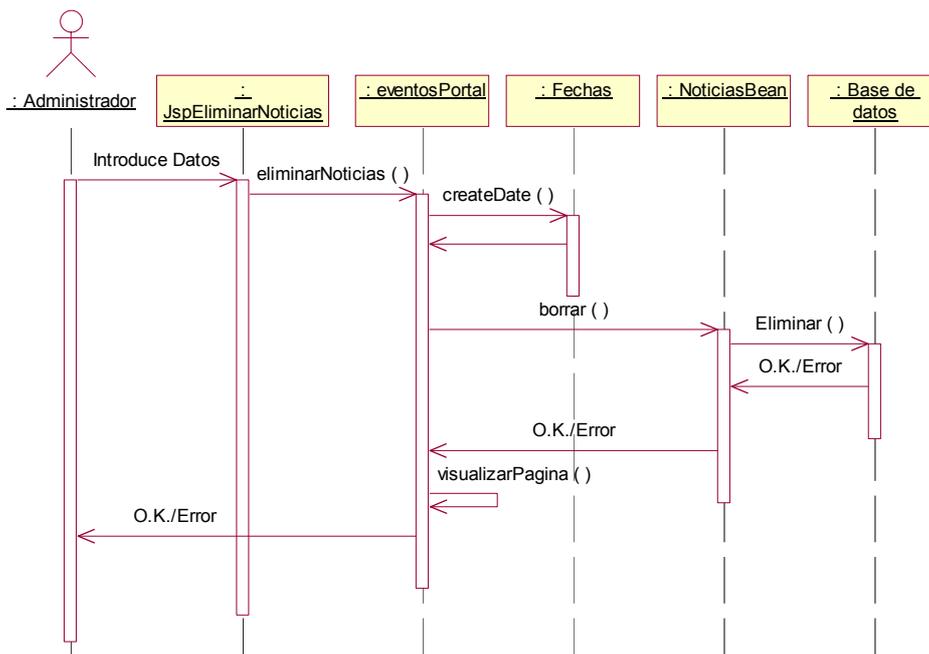
8.3 Eliminar Nota



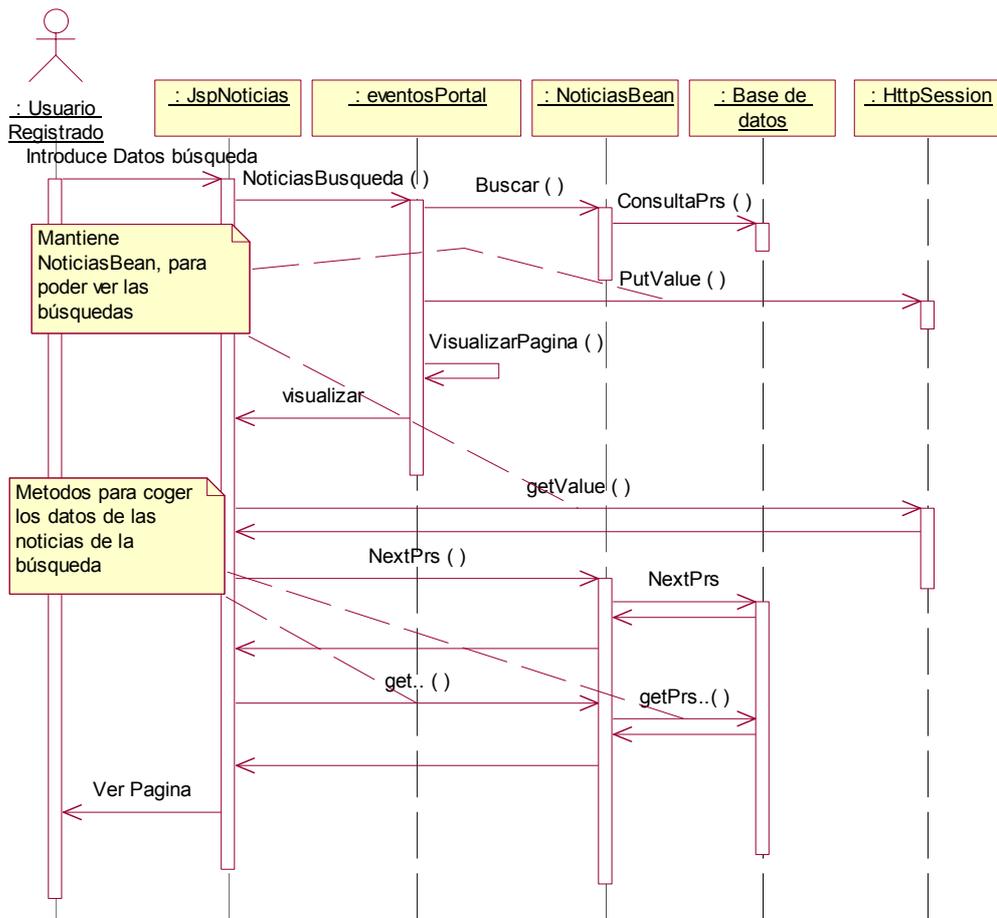
8.4 Insertar Noticia



8.5 Eliminar Noticia



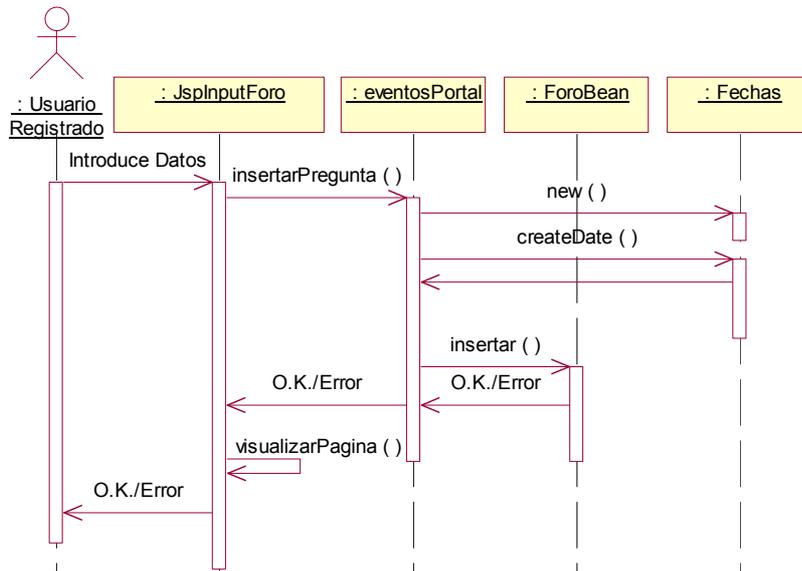
8.6 Leer Noticias



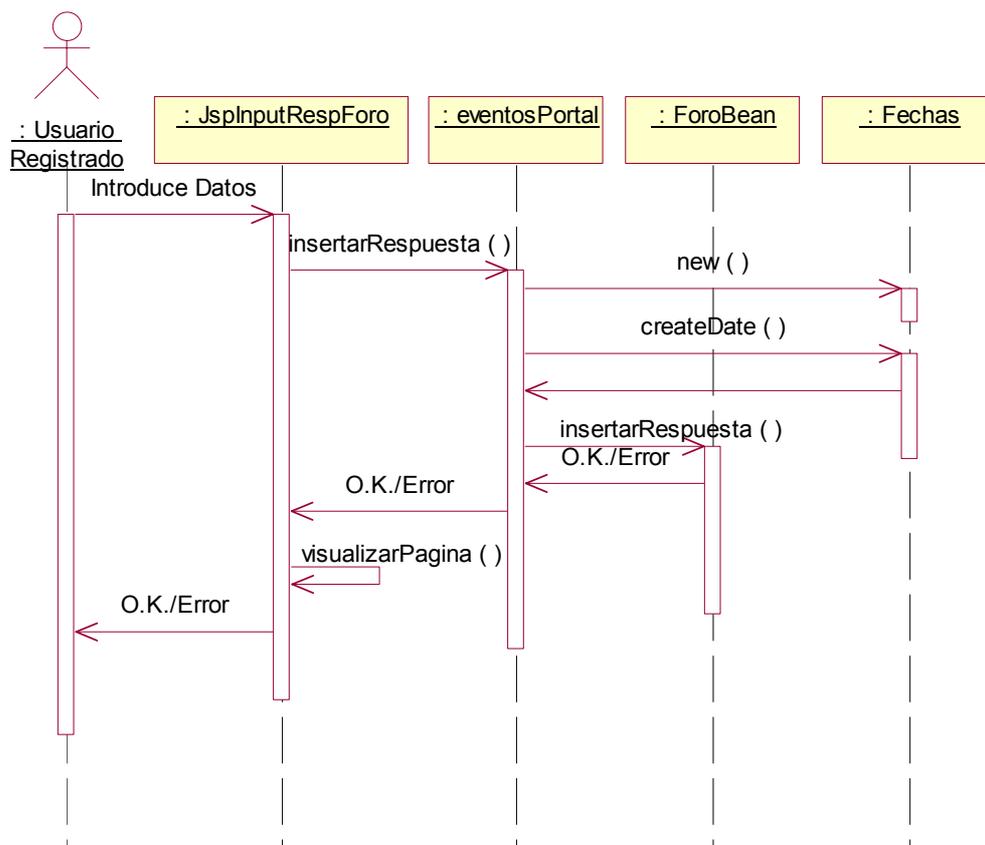
8.7. Insertar pregunta o respuesta en el foro

Diferenciaremos entre insertar una pregunta y una respuesta

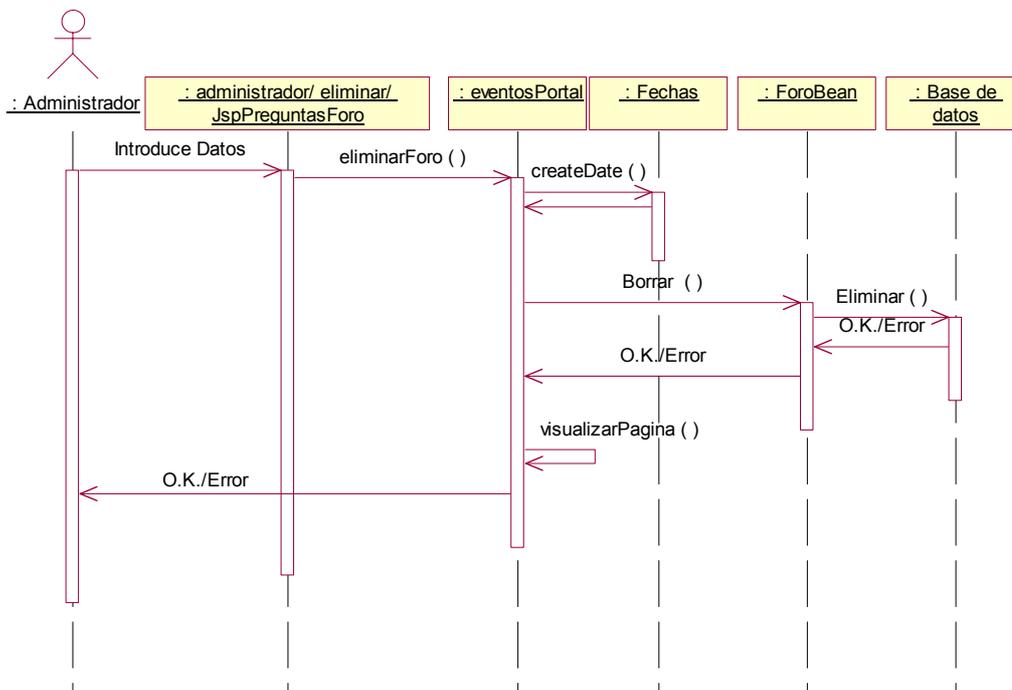
8.7.1. Insertar una pregunta



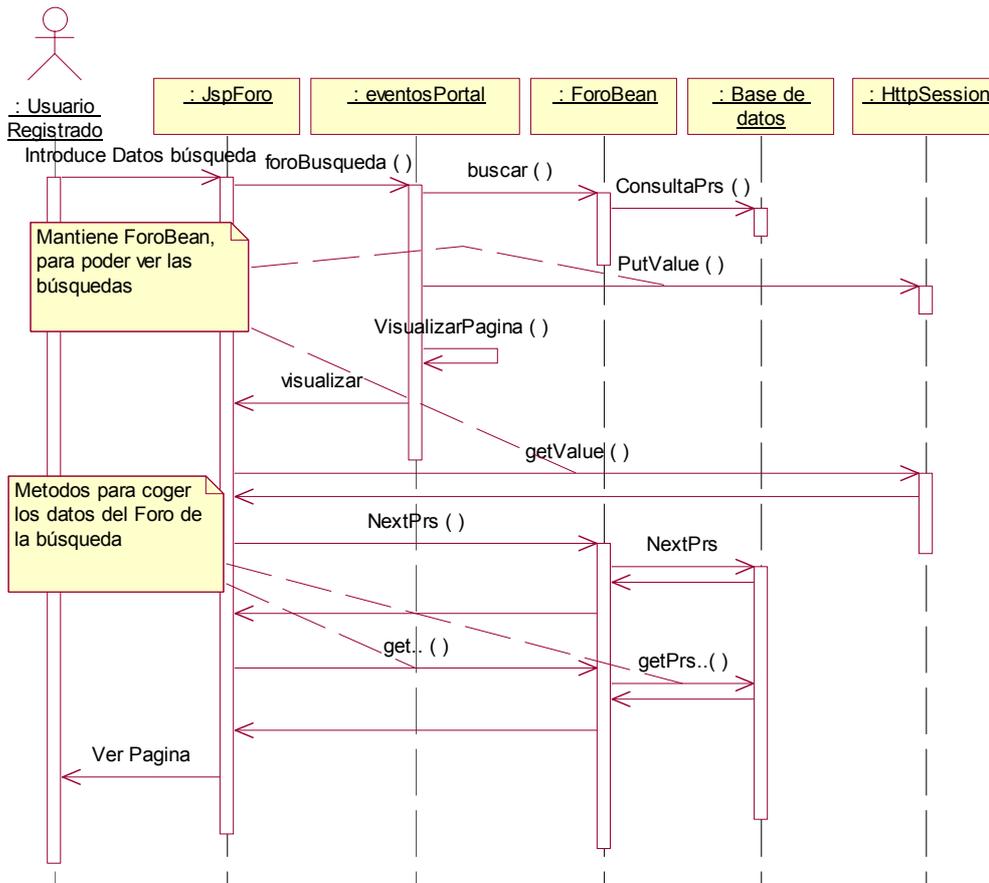
8.7.2. Insertar una respuesta



8.8 Borrar preguntas Foro



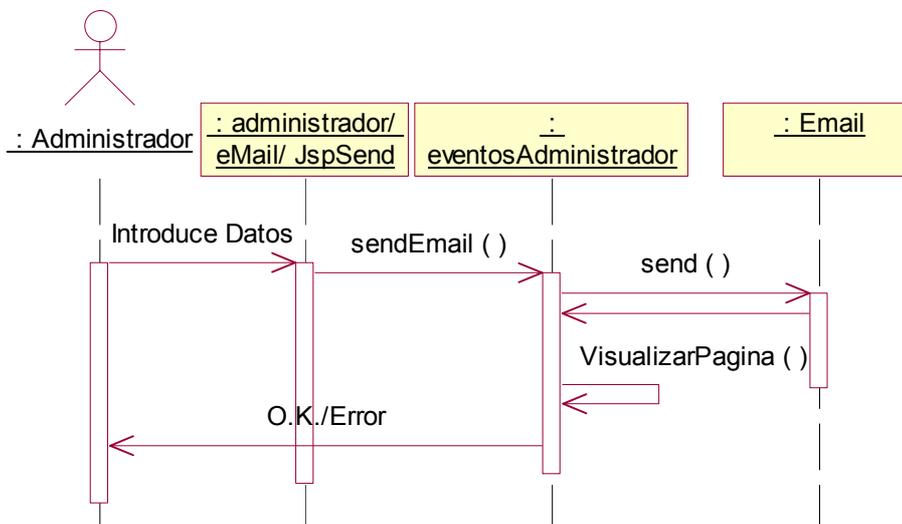
8.9 Ver preguntas o respuestas del foro



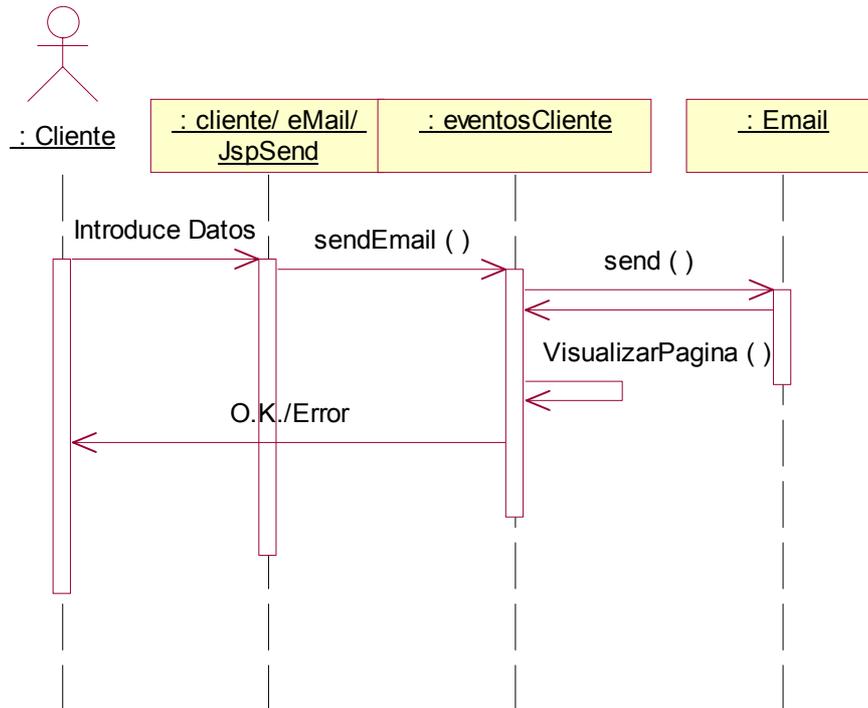
8.10 Enviar Correo Electrónico

Los dividiremos en tres diagramas de colaboración, dependiendo del usuario que lo inicie.

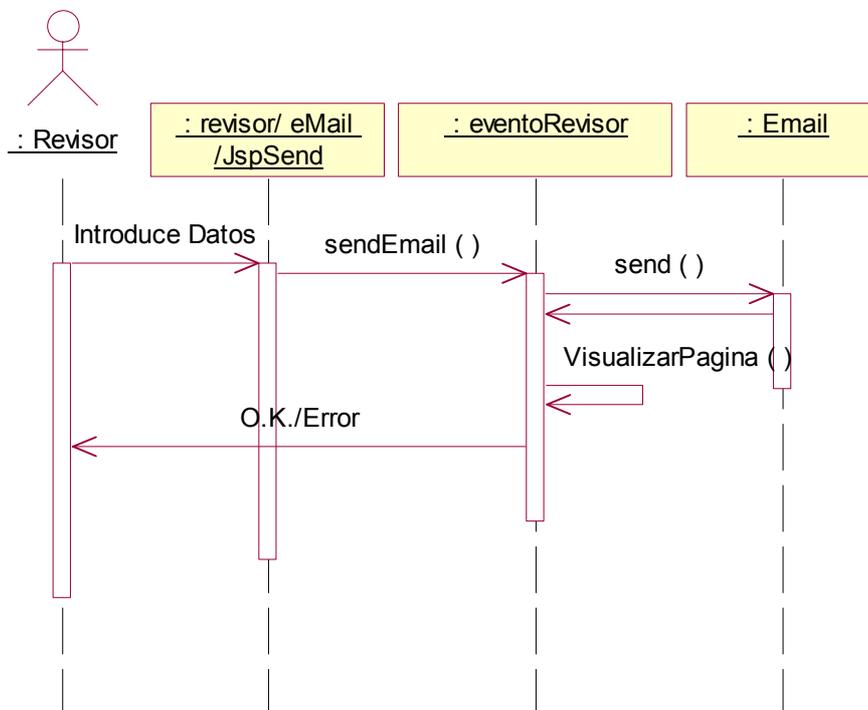
8.10.1 Iniciado por Administrador



8.10.2 Iniciado por cliente

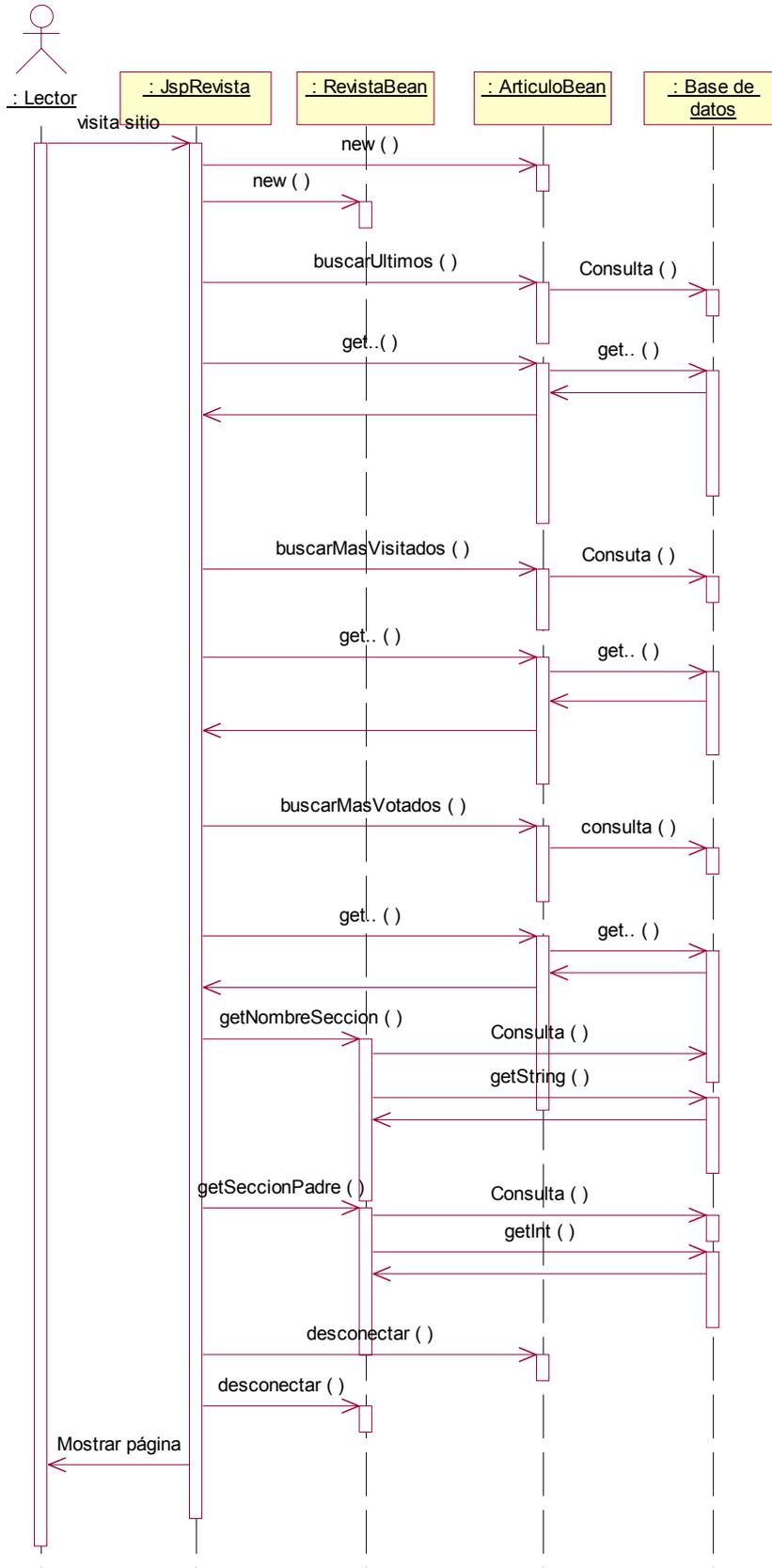


8.10.3 Iniciado por revisor

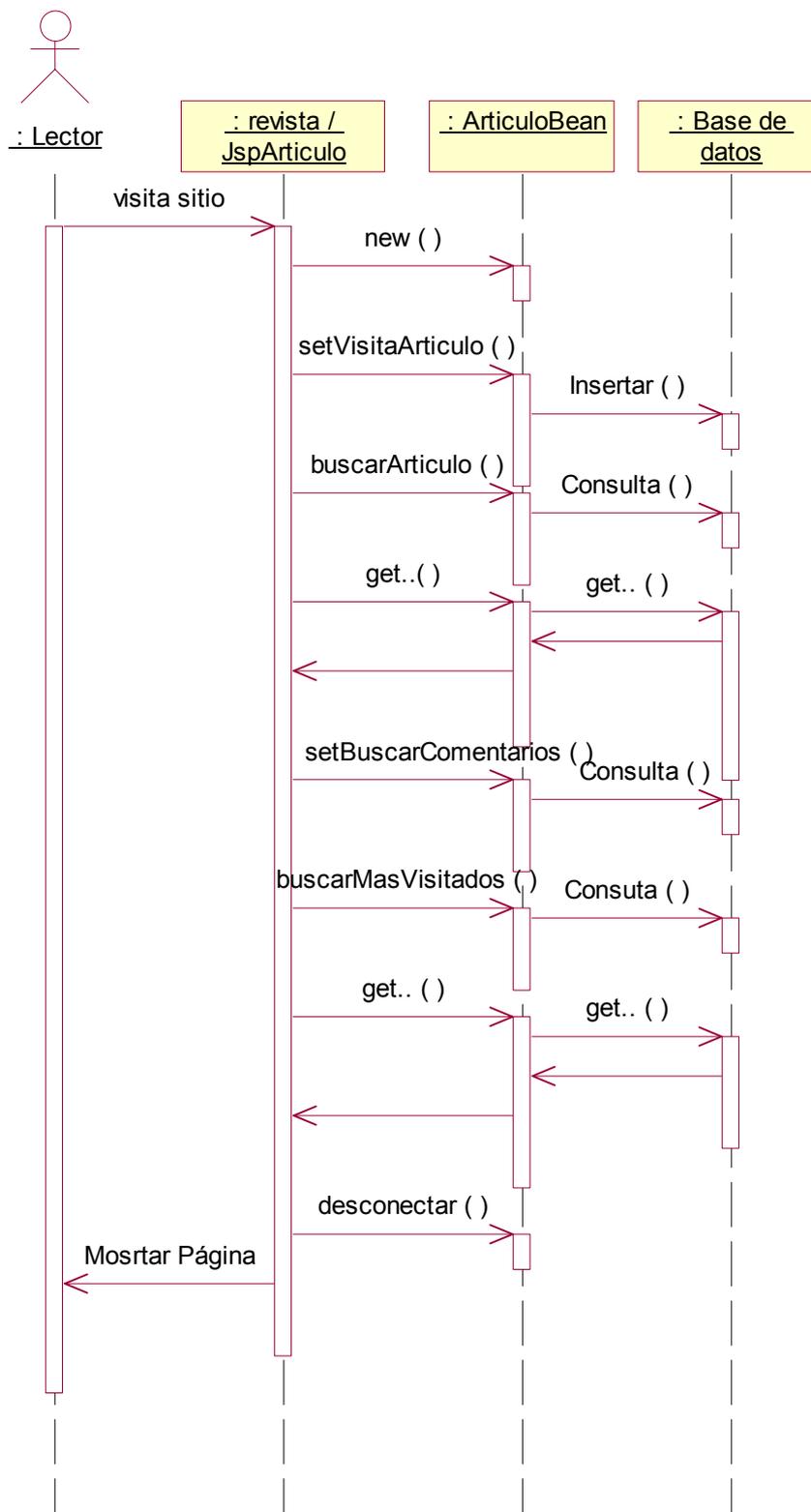


9) Consulta de la revista

9.1 Leer revista



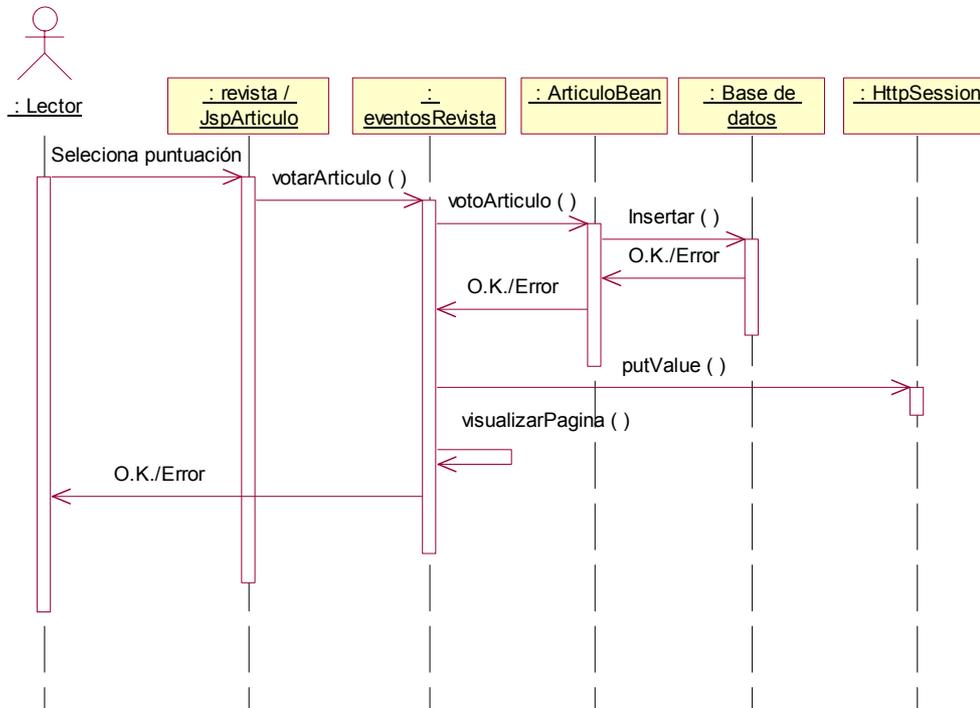
9.2 Leer artículo



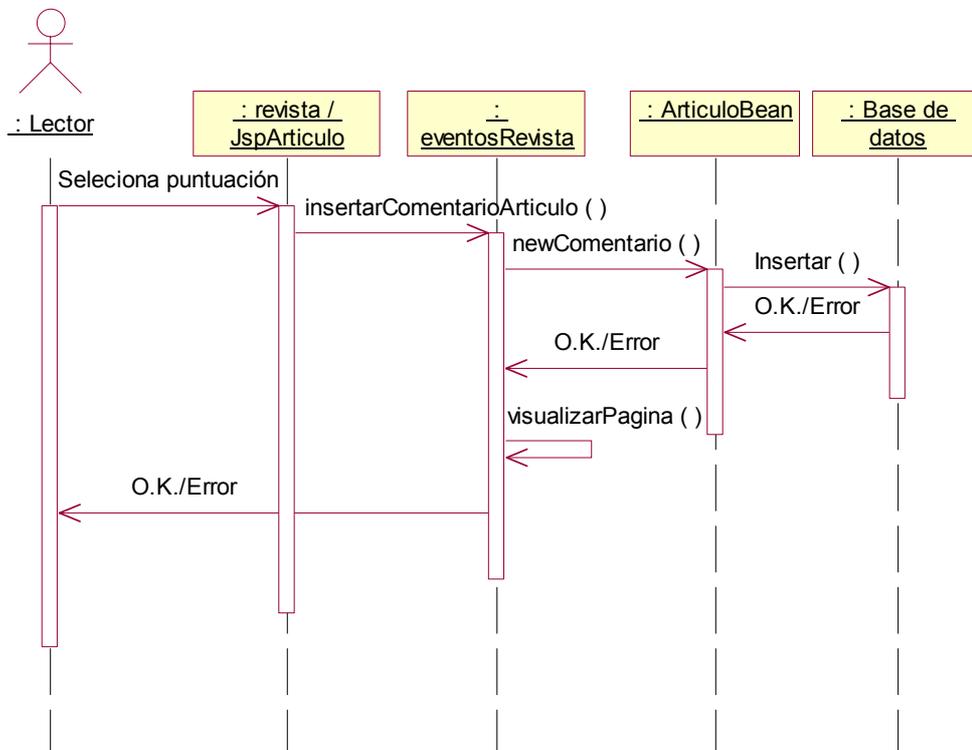
9.3 Interaccionar con artículo

Lo dividiremos en dos diagramas distintos: Votar artículo y comentar artículo.

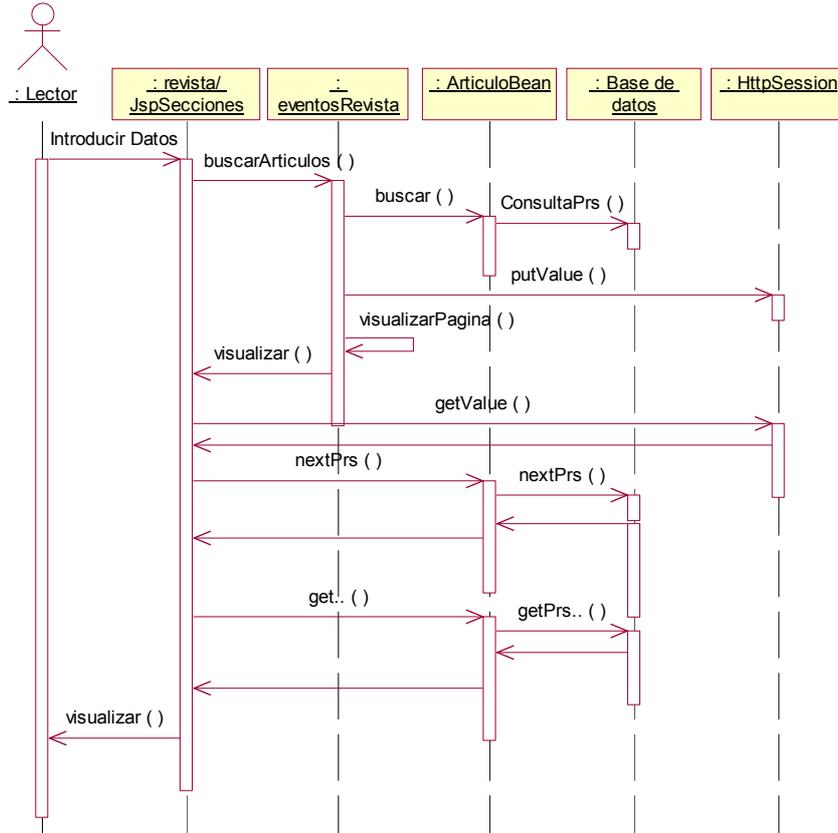
9.3.1 Votar artículo



9.3.2 Comentar artículo



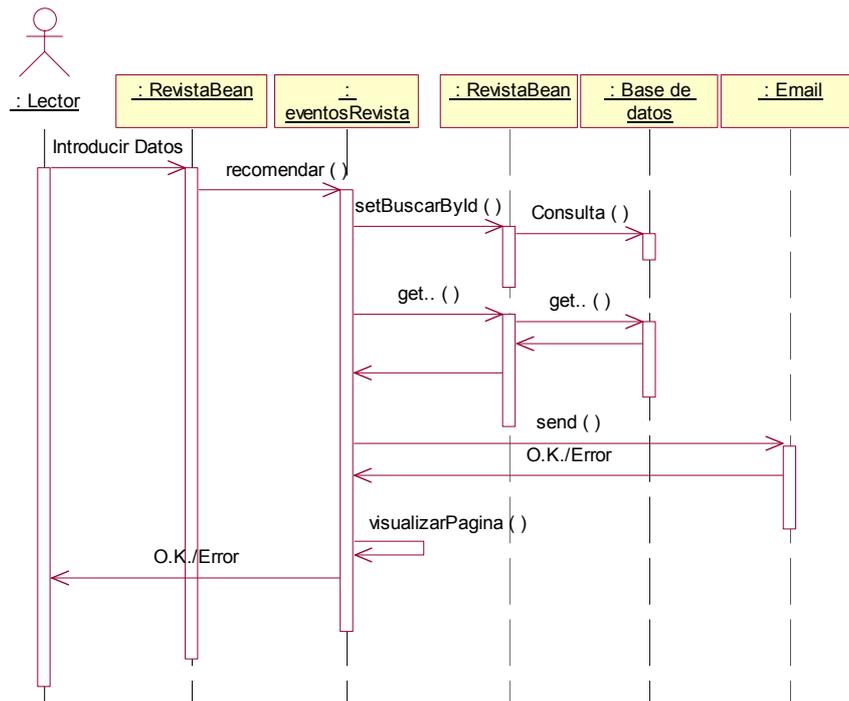
9.4 Buscar artículos



9.5 Establecer como página de inicio o agregar a favoritos

Estas operaciones son tan inmediatas que no hace falta la construcción del diagrama de colaboración para poder comprenderlas.

9.6 Recomendar la revista

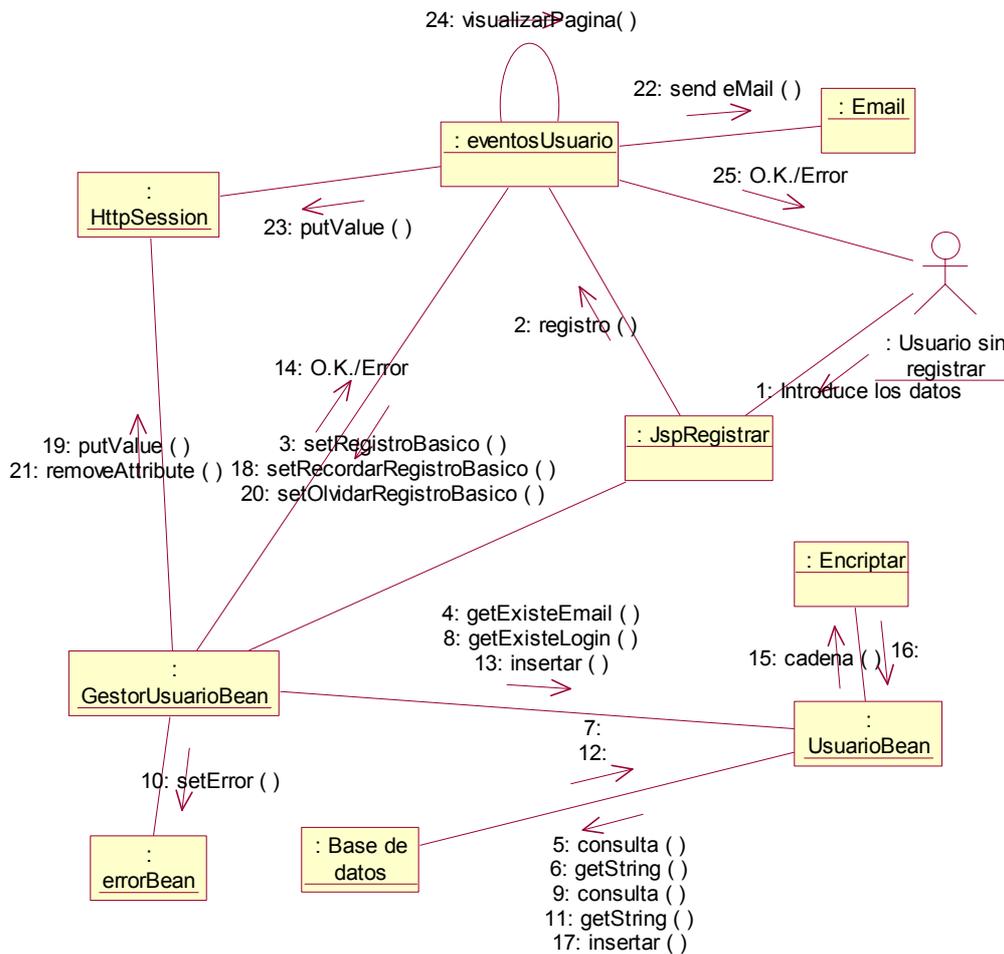


3.3.2 Diagramas de colaboración

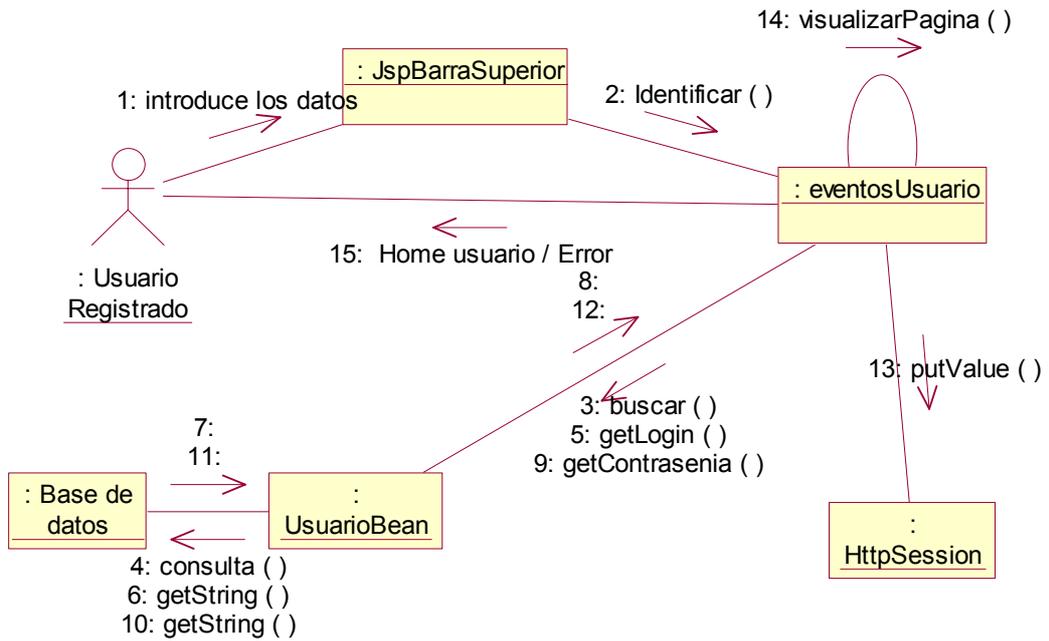
Captan una imagen del sistema en un determinado instante, destaca sobre todo la organización de los objetos que envían y reciben mensajes y nos permitirán profundizar sobre el nivel de detalle.

Se realizará diagramas de colaboración de los escenarios más interesantes de cada caso de uso. Se usarán en los casos en los que aporten más detalles a los aportados por los diagramas de secuencia.

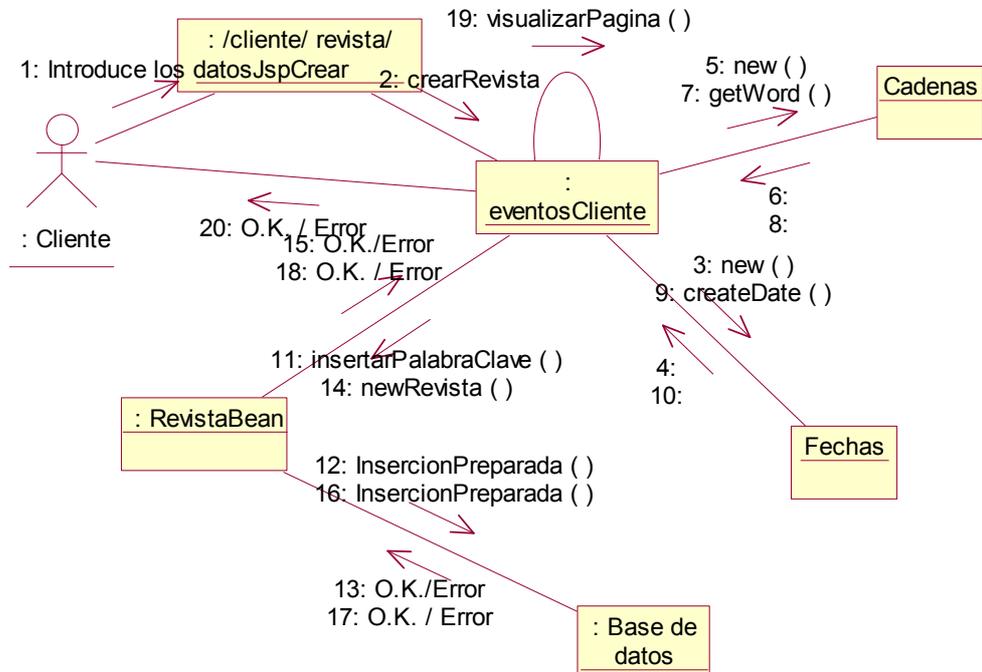
1.1 Registro de Usuarios



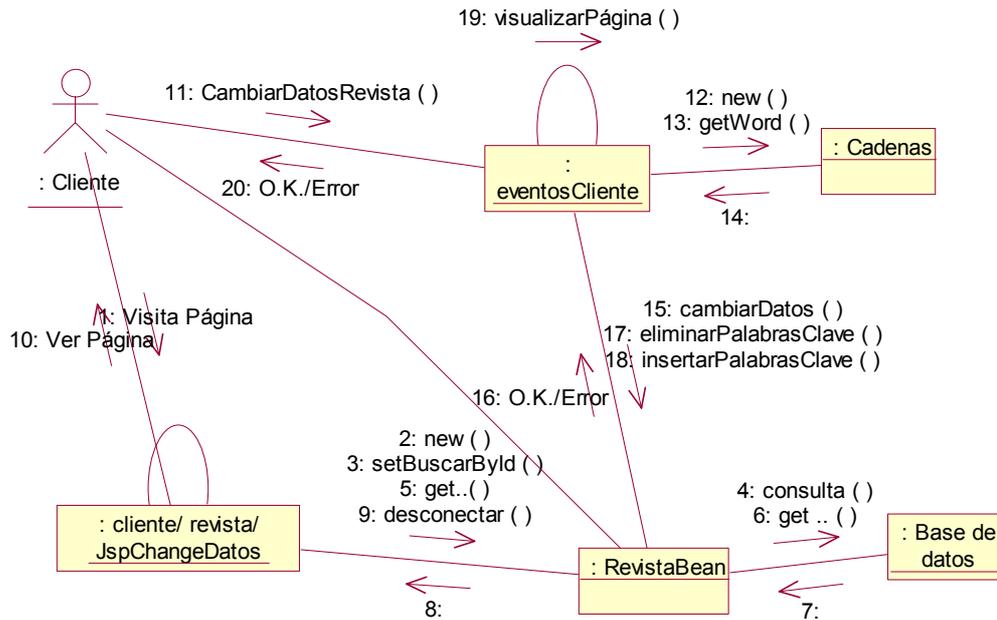
1.2 Validar usuarios



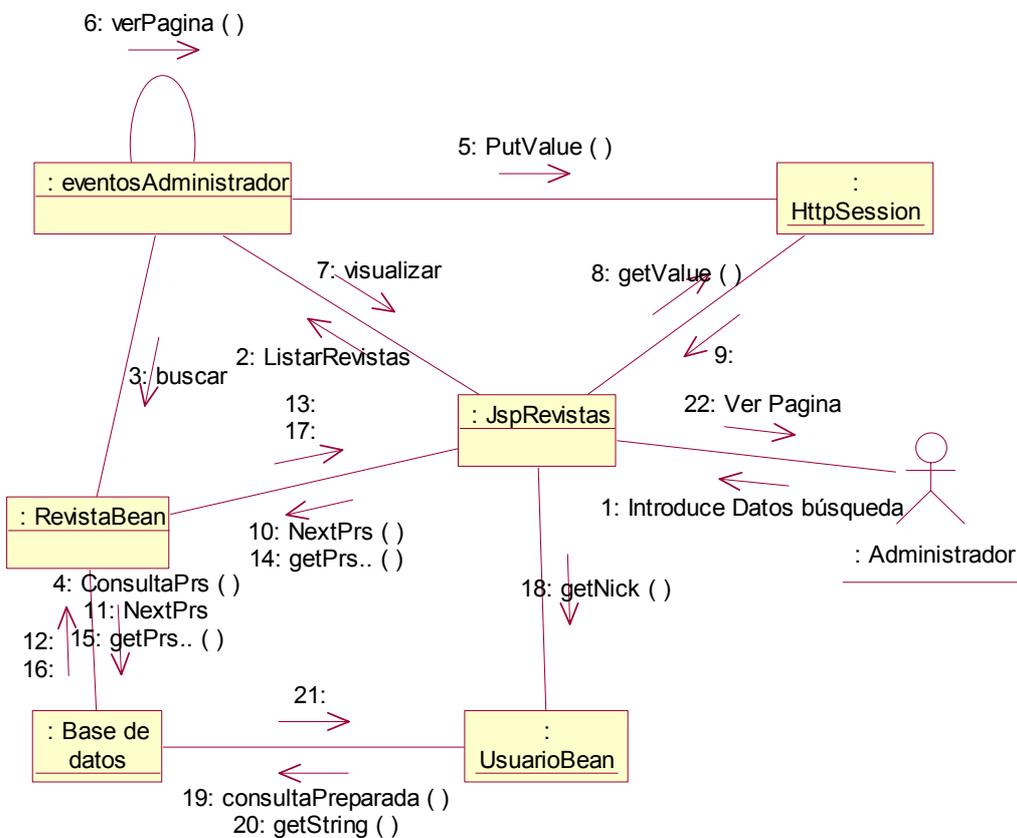
4.2 Crear una revista



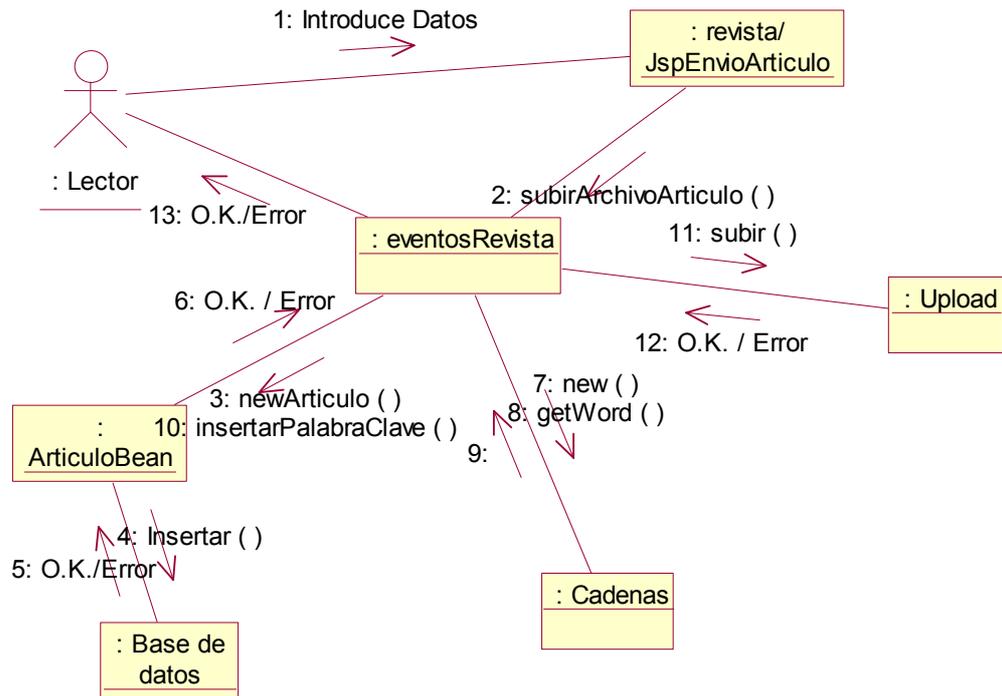
4.3 Modificar una revista



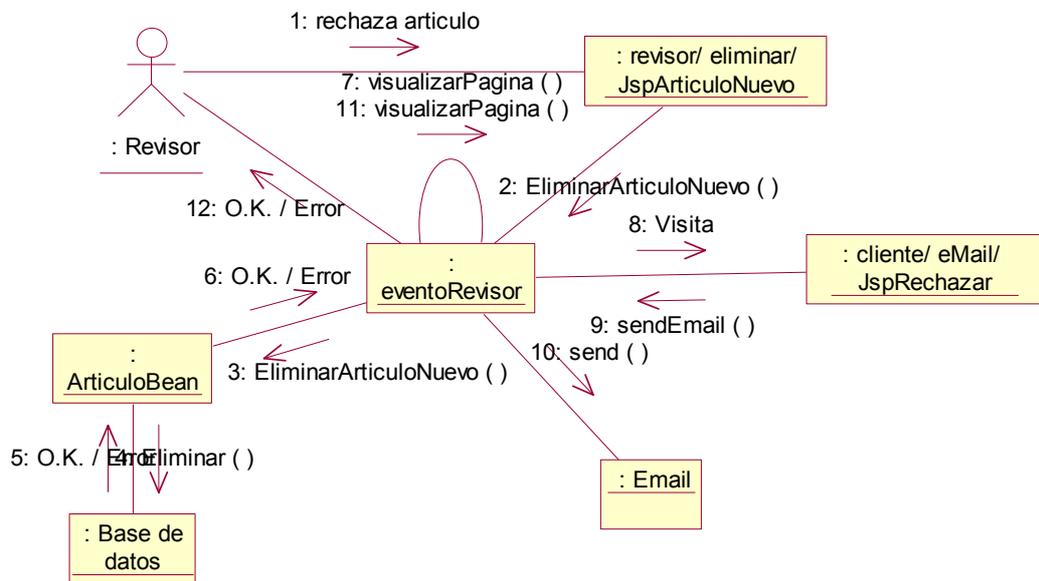
4.10 Listar revistas



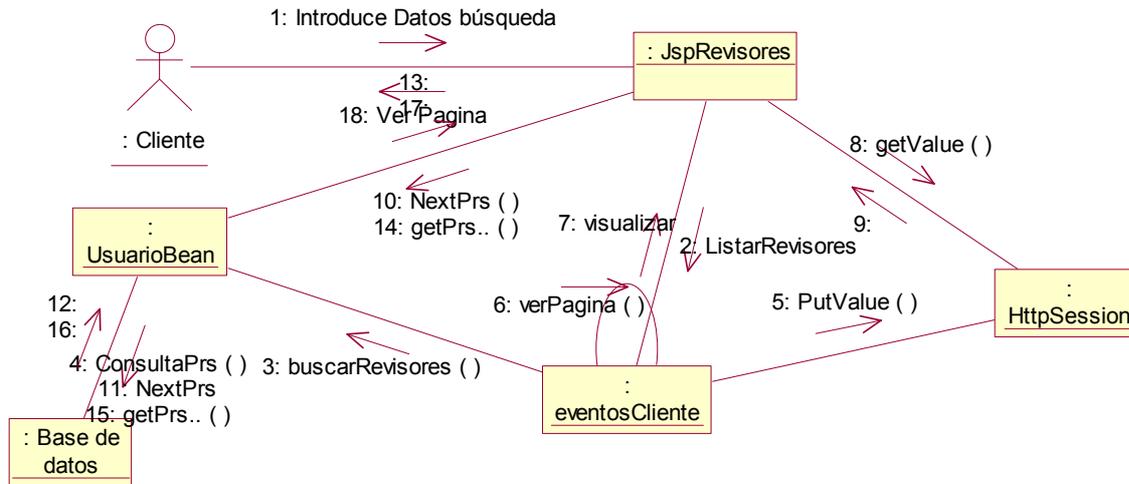
5.1 Enviar artículo



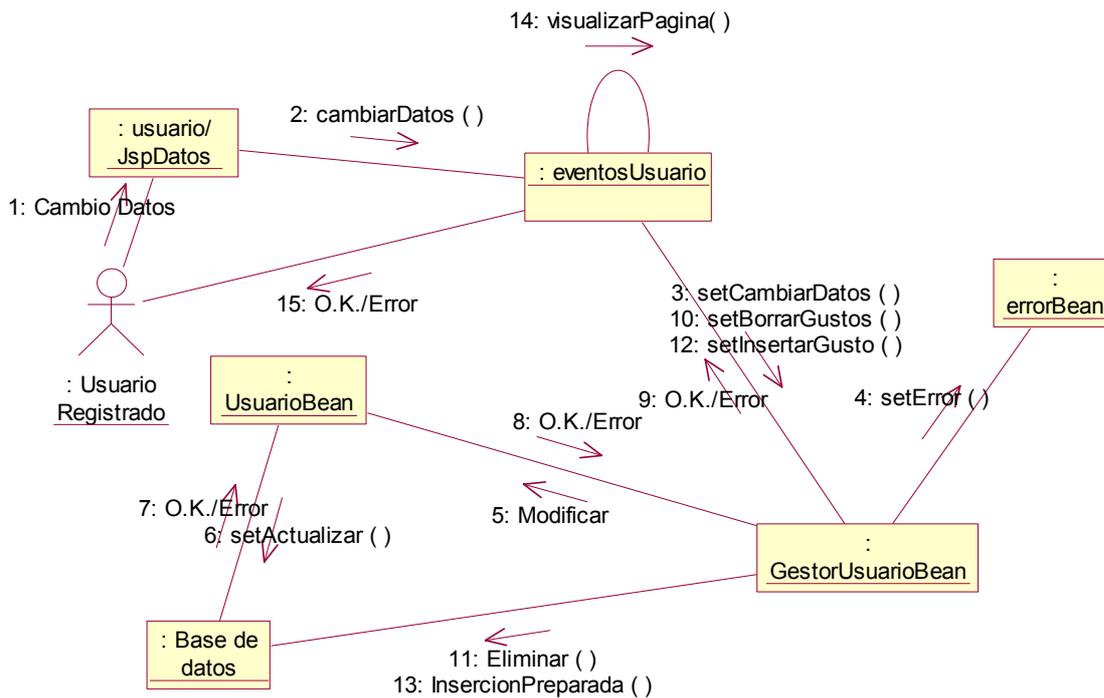
5.6 Rechazar artículo



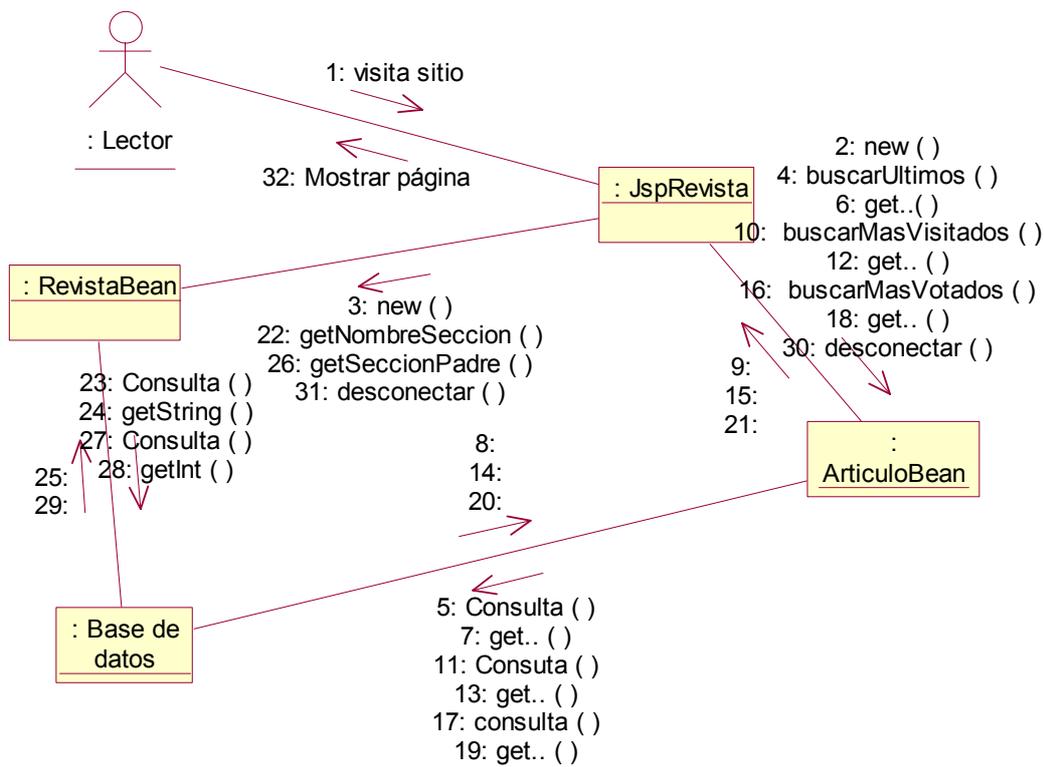
6.1.2 Listado de usuarios (Iniciado por Cliente)



7.1 Modificar datos de un usuario



9.1 Leer revista



3.4 Diagramas Estáticos

A lo largo del diseño se han ido perfilando las clases definitivas a emplear en la implementación de la herramienta, gracias a las ideas surgidas en el análisis y añadiendo aquellas que fueran necesarias para estructurar y mantener la lógica de la herramienta.

Los diagramas estáticos (o de clases en la metodología Booch) se utilizan para modelar la vista de diseño estática de un sistema. Son los más importantes del DOO (diseño orientado a objetos), y son la piedra angular de nuestro diseño. Contienen toda la información de todas las clases y sus relaciones con otras. Aunque son los más importantes, no se llega a ellos directamente dado que tienen un gran nivel de abstracción, para lo cual hay que contemplar el modelo globalmente sin particularizarse en ningún escenario en concreto. Se debe tener en cuenta que el diagrama de clases se refinará en las sucesivas iteraciones del modelo.

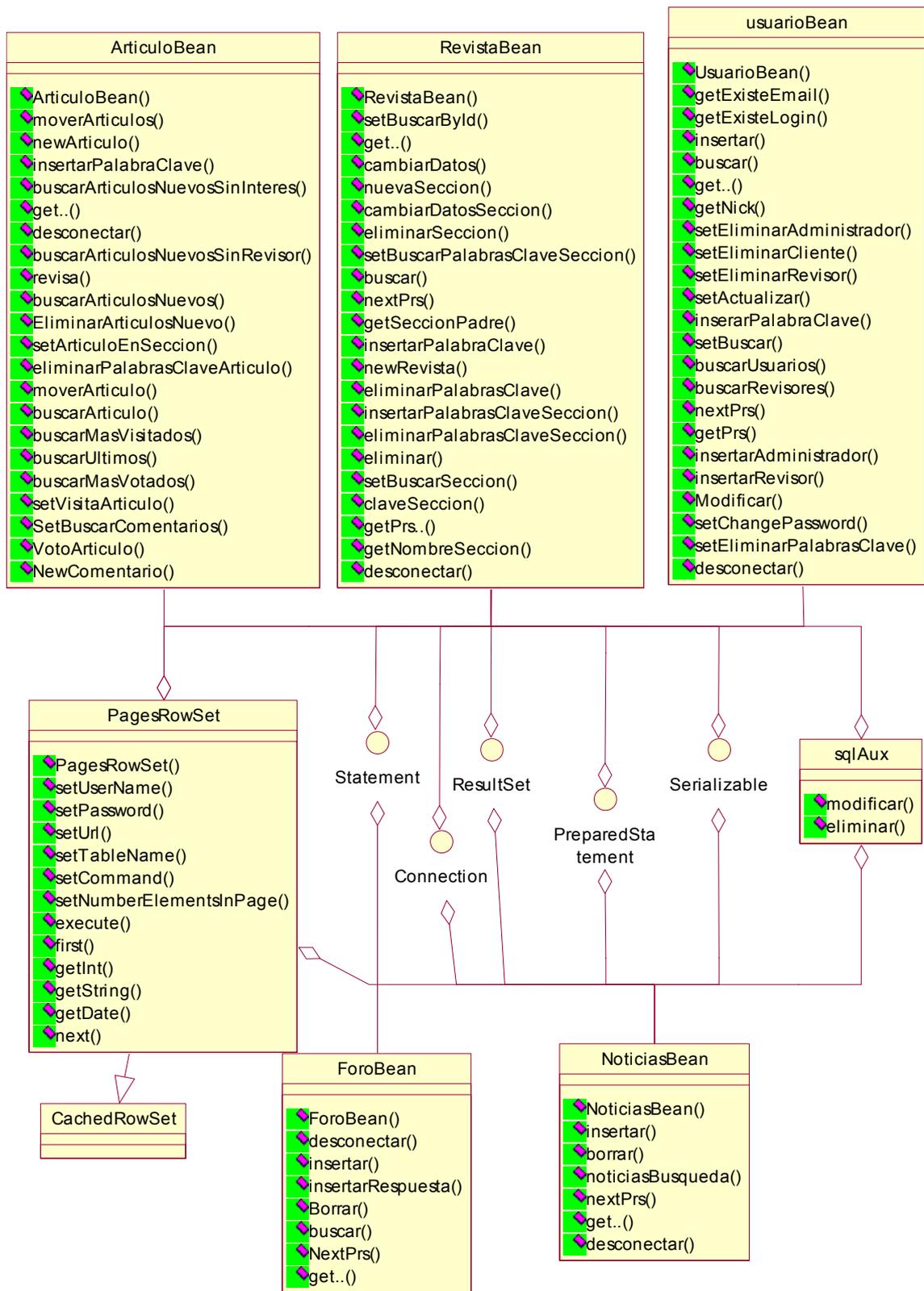
Así, vamos a ver a continuación las diferentes relaciones entre las clases escogidas para modelar cada una de las operaciones básicas que vienen a componer la herramienta.

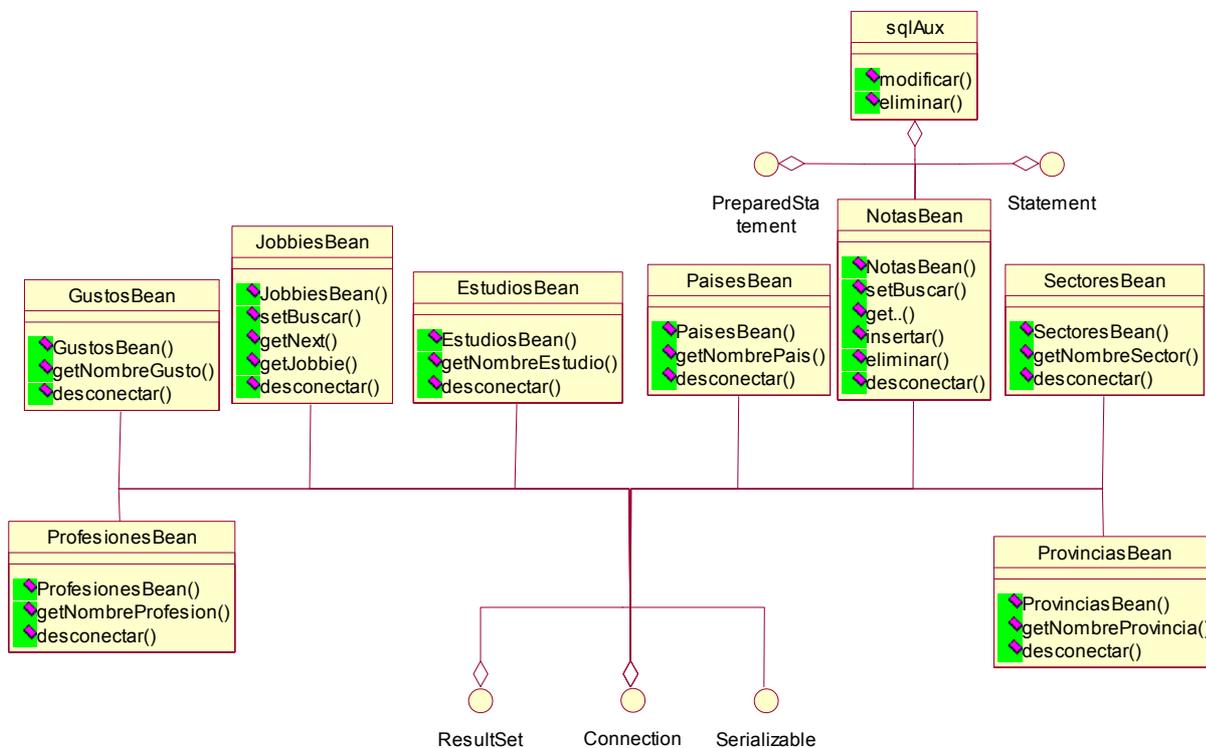
Para ver los diagramas estáticos nos centraremos en el patrón Modelo-Vista-Controlador (MVC) que conformarán nuestra aplicación. Para empezar nos fijaremos en el Modelo y posteriormente nos fijarnos en la relación que existe entre las clases que forman parte del modelo con las clases controlan la aplicación.

3.4.1 Diagramas de clases para el Modelo (Acceso a los Datos)

Estos diagramas que se muestran un poco más abajo, vienen a representar las clases que modelarían el acceso a la información que esté contenida en la base de datos de la aplicación y que también nos proporciona los medios para modificarlos.

Se ha dividido en dos el diagrama para facilitar la comprensión por parte de lector y para organizar el esquema dentro de esta documentación.



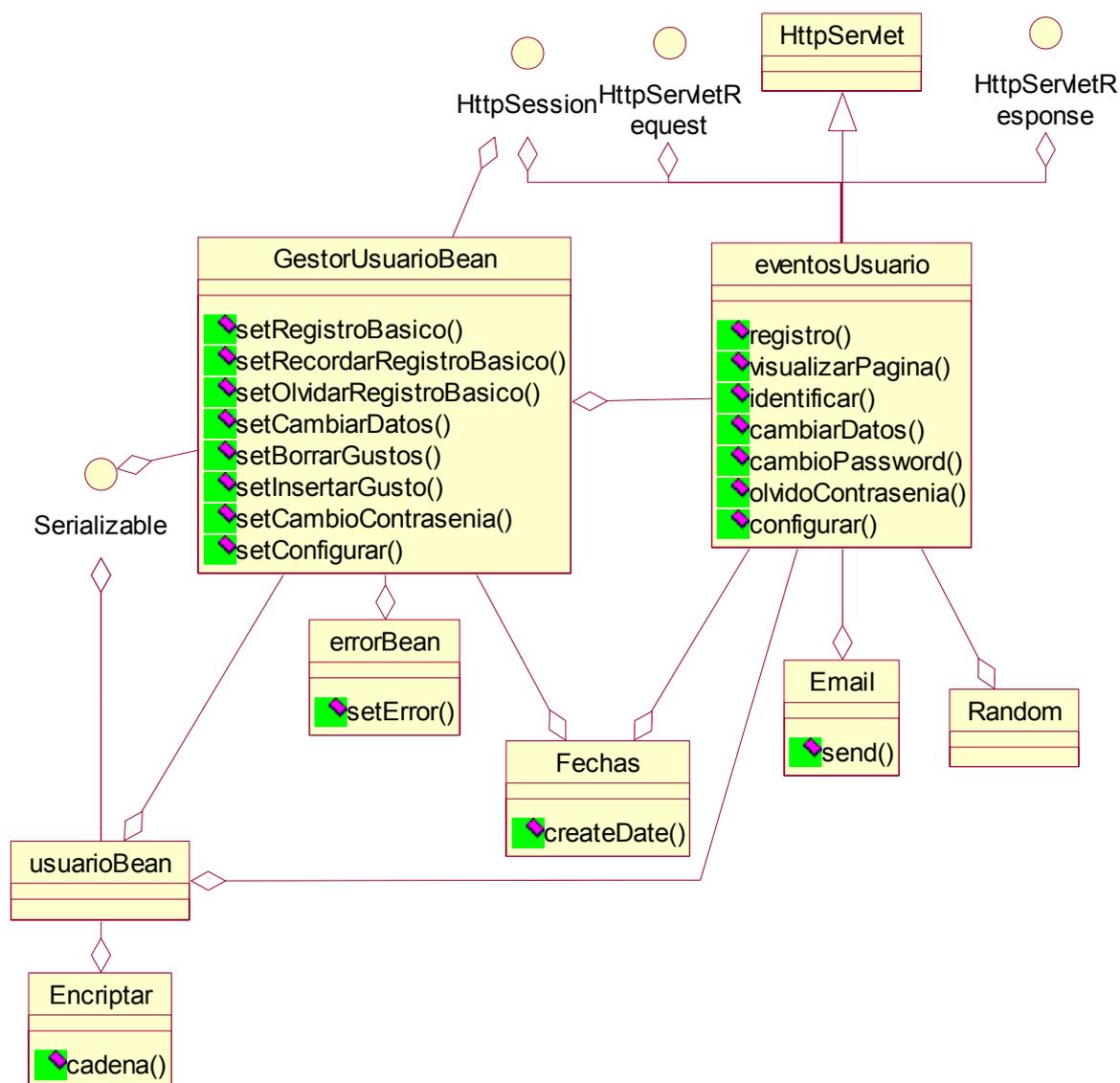


3.4.2 Diagrama de clases para el Modelo-Controlador

En este apartado nos centremos en los controladores y las relaciones que tienen con las clases que forma parte del Modelo dentro del patrón MVC.

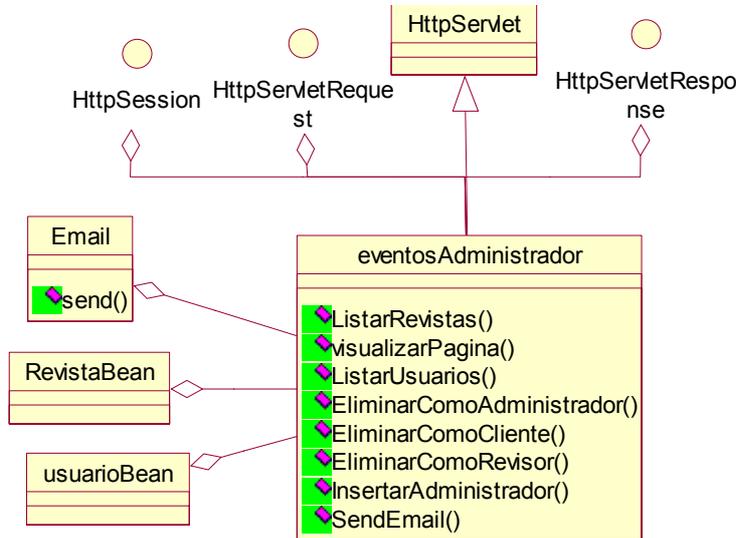
3.4.2.1 Controlador eventos Usuario

Este captura los eventos de un usuario registrado y los procesa para realizar las acciones pertinentes y pasar al modelo los datos que necesita o quiere almacenar en la base de datos.



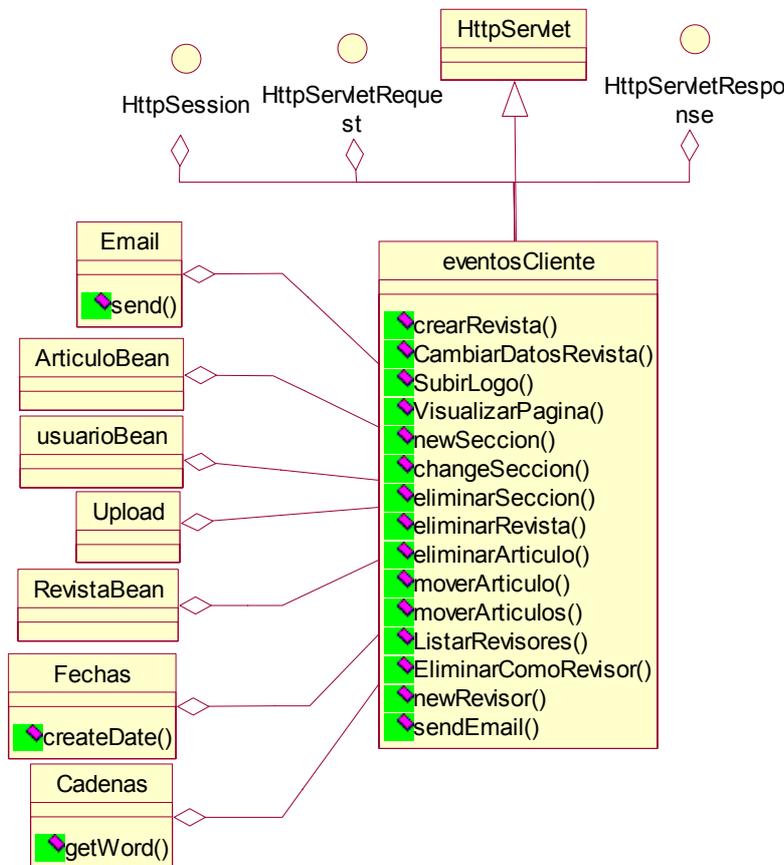
3.4.2.2 Controlador eventos Administrador

En este caso es el encargado de capturar la mayoría de los eventos que puede efectuar un Administrador y los procesa introduciendo la información en la base de datos a través del Modelo, visto con anterioridad.



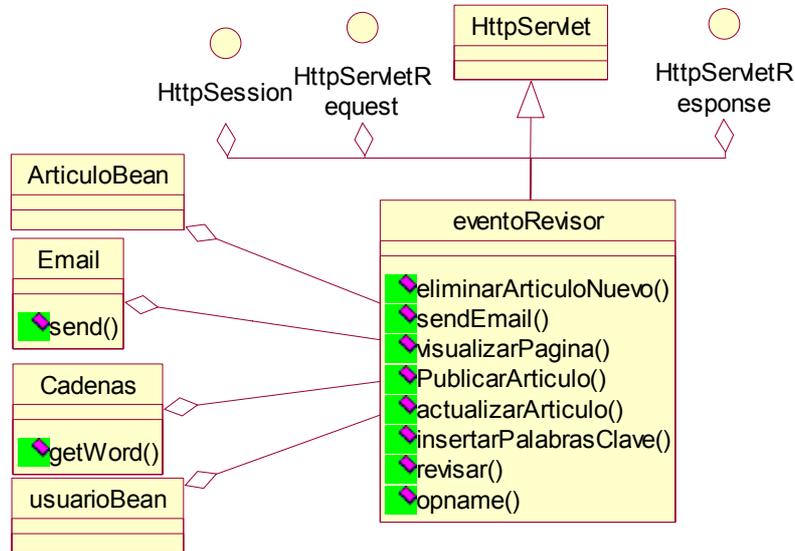
3.4.2.3 Controlador eventos Cliente

De igual manera que el anterior, este se encarga de capturar y procesar la mayoría de los eventos que puede efectuar un usuario con privilegios de cliente.



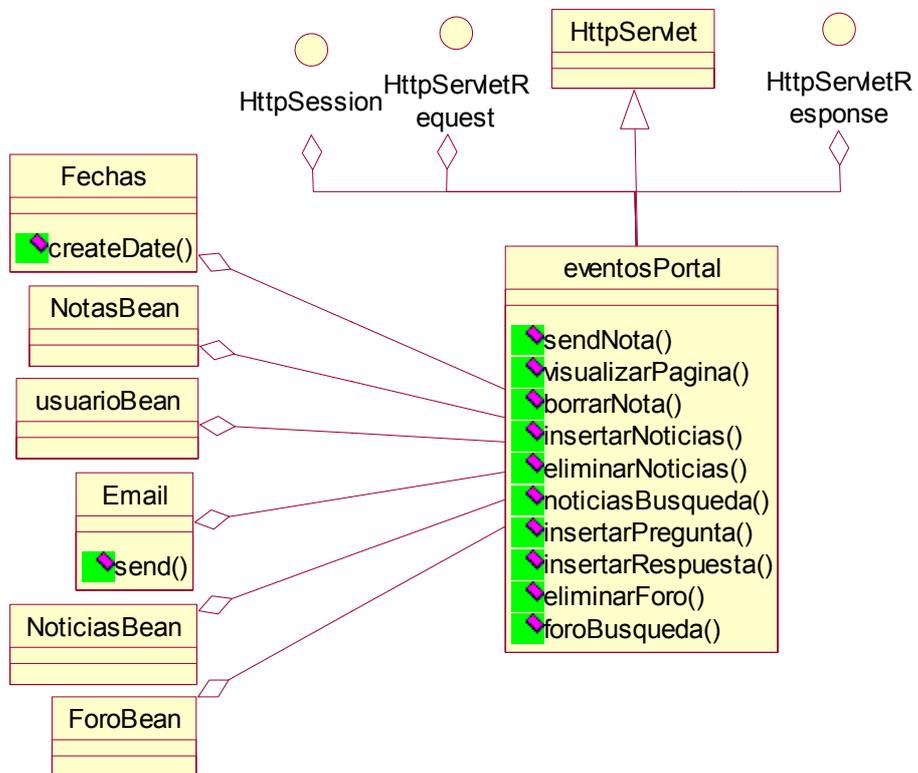
3.4.2.4 Controlador eventos Revisor

Al igual que los anteriores, se encarga de capturar y procesar la mayoría de los eventos que puede efectuar un usuario que en este caso tendrá privilegios como revisor de alguna de las revistas que hay en el sistema.



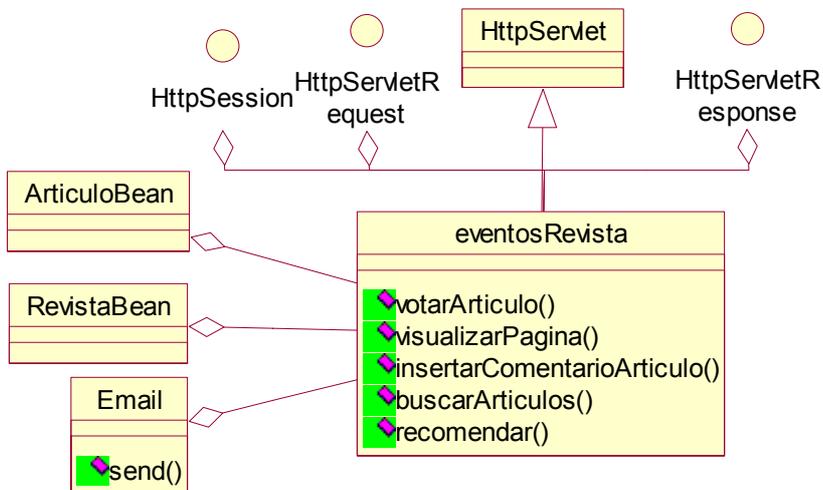
3.4.2.5 Controlador eventos Portal

A diferencia de los anteriores, no sólo se encarga de capturar los eventos de un usuario con un determinado privilegio, si no que captura los eventos que pueden hacer los distintos usuarios para establecer una comunicación entre ellos: Envío de notas, noticias y preguntas-respuestas en el Foro.



3.4.2.6 Controlador eventos Revista

Este controla los eventos que se pueden producir al visitar alguna de las revistas que nuestra herramienta genera.



3.5 Diagramas de Actividad

Estos diagramas vienen a representar un flujo de datos, es decir, los distintos pasos que se han de llevar a cabo para realizar una determinada tarea. Estos no son otra cosa que grafos que parten de un estado inicial y que se va cambiando de actividad a través de transiciones automáticas, así cuando una actividad es terminada se pasa automáticamente a la siguiente.

Estos nos ayudarán a comprender cuales son los pasos que conlleva realizar una determinada operación sin necesidad de conocer cuales son los mensajes internos que se pasan los distintos objetos.

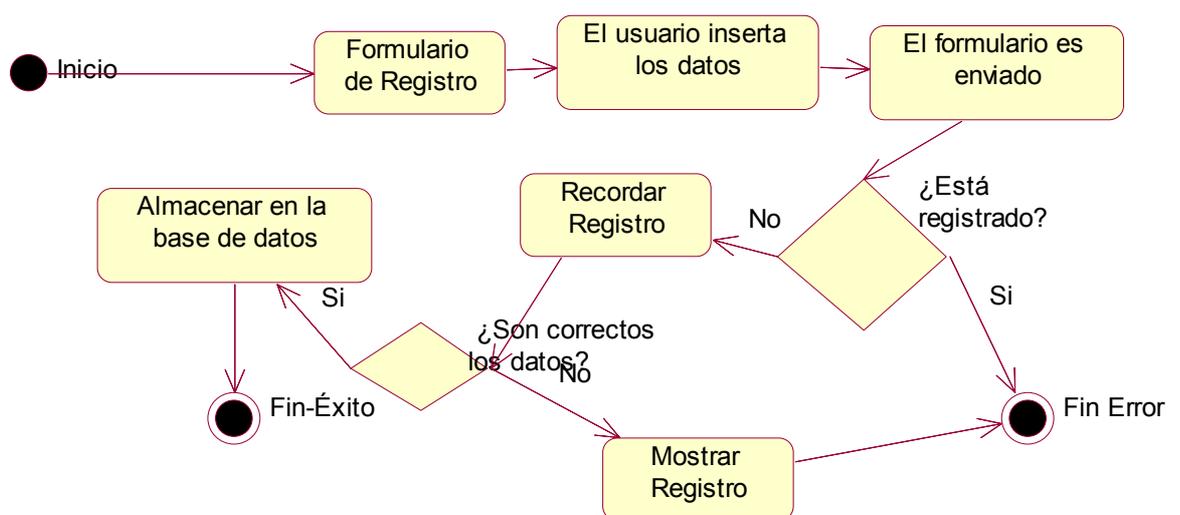
A diferencia de los diagramas de estados, aquí cada actividad no espera a que suceda un evento para pasar al siguiente paso, sino que la actividad permanecerá activa mientras dure su ejecución, y una vez finalizada, la siguiente actividad será la que se active.

Estos diagramas pueden contener bifurcaciones, estas se dan cuando dependiendo del resultado de una actividad será necesario realizar una u otra actividad. Las bifurcaciones son divisiones del flujo de actividades en los que la ejecución toma un camino u otro en función del resultado de la actividad anterior.

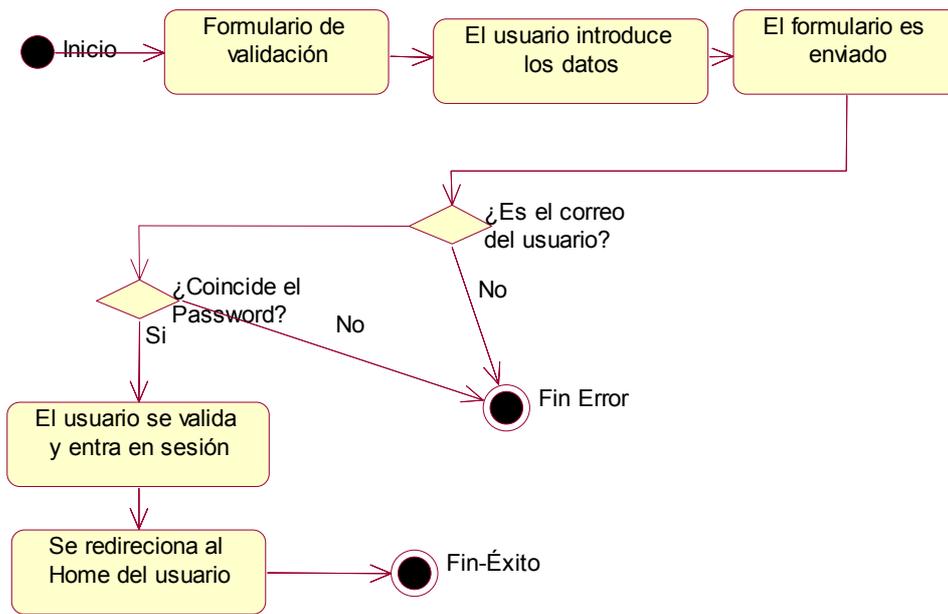
Debido a que estos diagramas vienen a aclarar los pasos necesarios que se han de dar para llevar a cabo una determinada operación, sólo veremos aquí los diagramas de actividad correspondientes a aquellas operaciones que realmente tengan un flujo de actividades y ejecuciones un poco más complicadas de lo normal, y que no se hayan visto aclaradas por otros diagramas o con otras explicaciones. Habrá otros pasos que aun siendo un poco complicados se repetirán en varias operaciones por tanto se explicará en el primero y el resto se obviará.

Se enumerarán estos diagramas asociándolos con los escenarios del análisis.

1.1 Registro Usuario

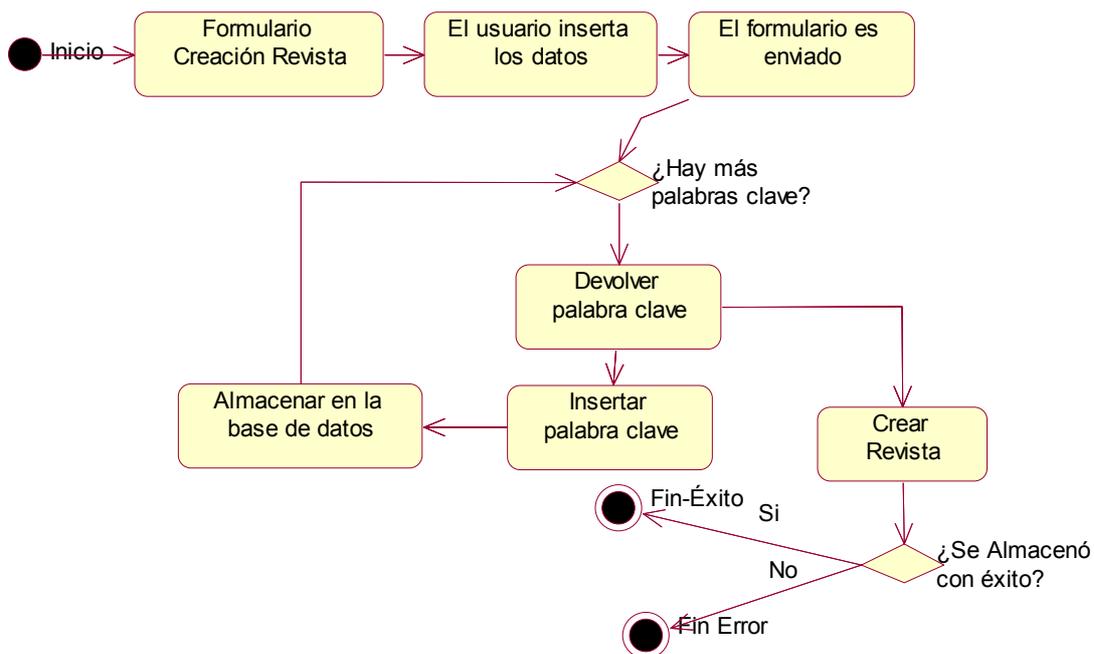


1.2 Validación usuario

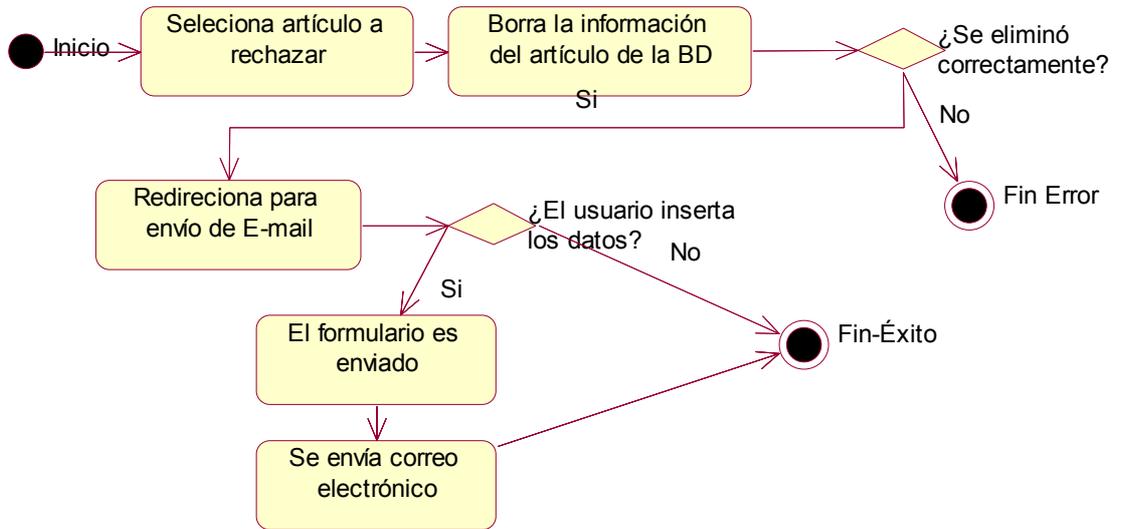


4.2 Crear una Revista

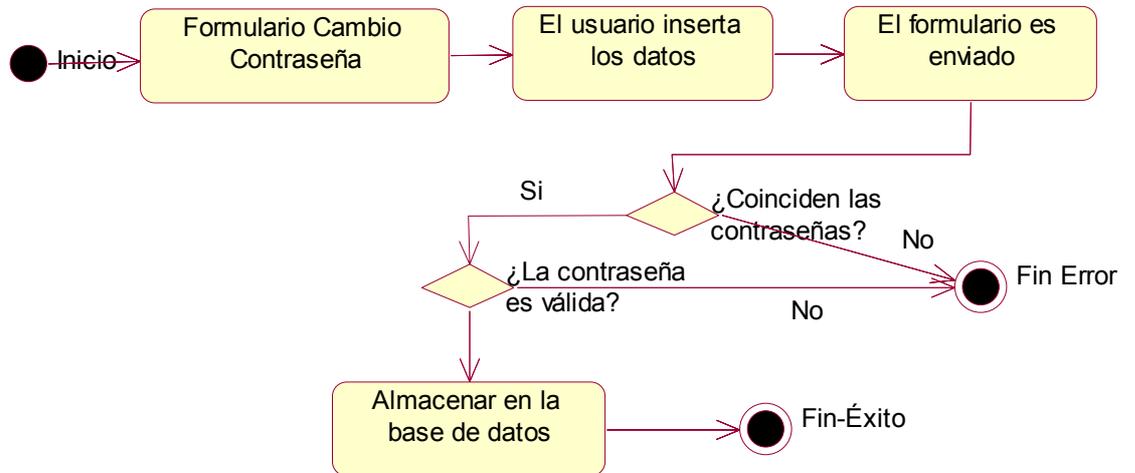
Este diagrama representa un flujo de datos muy interesante que se repite en multitud de escenarios, es el referente a las palabras clave. Este es un ciclo que va obteniendo las palabras clave del formulario enviado y posteriormente se van introduciendo en la base de datos, según corresponda. Un flujo de datos muy similar se repetirá cada vez que se introduce en la base de datos las palabras claves asociadas a un revisor, revista, artículo, etc. Por tanto el resto de diagramas no se expondrán porque es un flujo muy sencillo y queda perfectamente explicado en este diagrama.



5.6 Rechazar artículo

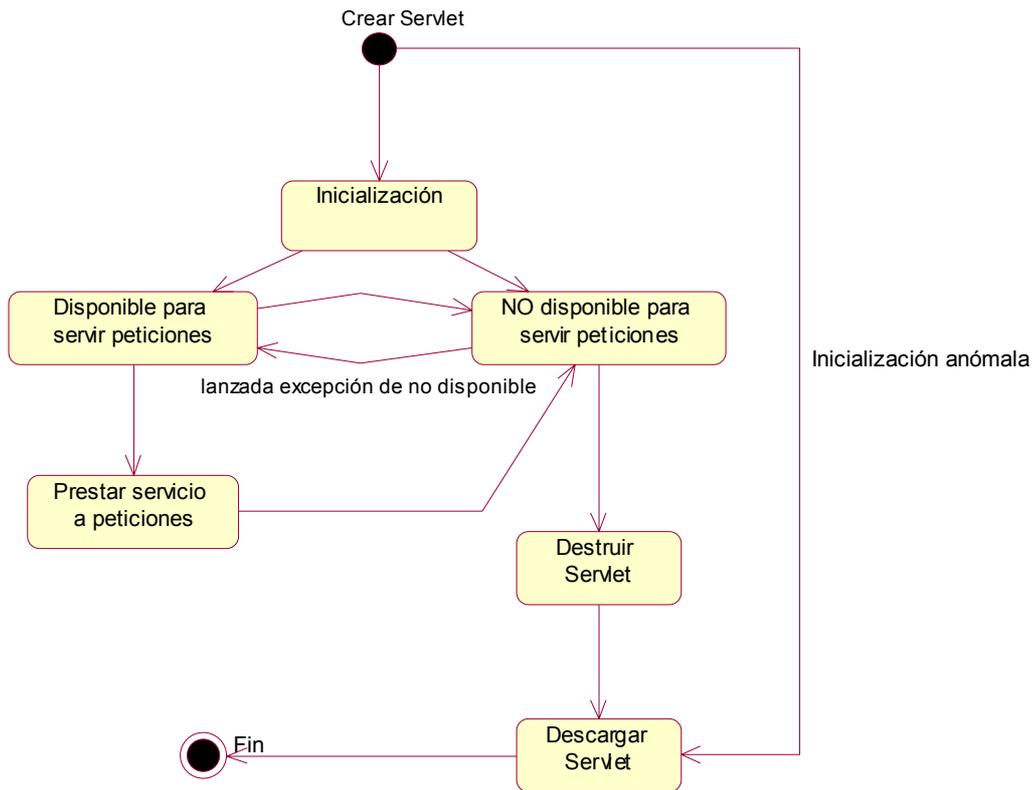


7.2 Cambiar Contraseña



3.6 Diagramas de estado

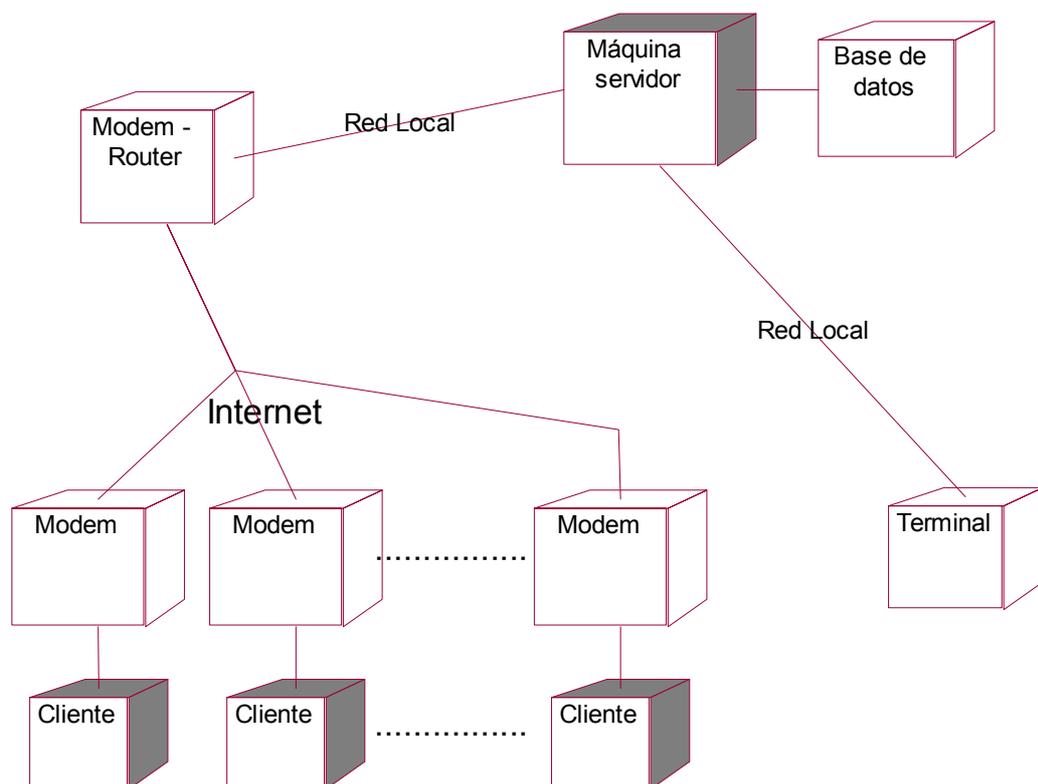
Estos diagramas se utilizan para modelar aspectos dinámicos del sistema intentando modelar los distintos estados por lo que pasa un objeto durante su vida correspondiéndose con los estímulos recibidos, sus respuestas y acciones, es decir, no sólo muestra los estados, si no los eventos que provocan una transición de un estado a otro. Debido a que cada diagrama de estados se corresponde con una clase y todos los servlets tienen en el mismo ciclo de vida, a continuación veremos un único diagrama de estados para un servlet cualquiera que se puede extrapolar a todos los servlets.



3.7 Diagramas de despliegue

Este es otro diagrama muy ligado a la implementación, este representa la topología hardware del sistema sobre el que funcionará nuestra aplicación o lo que es lo mismo, la disposición física de los distintos elementos que componen el sistema que se está desarrollando y en el que se indican cuáles son los componentes que conforman las distintas partes en las que nuestro sistema puede dividirse.

A continuación veremos el diagrama que viene a representar la estructura de nuestro sistema, donde cada nodo representado en el diagrama viene a ser un recurso computacional. Como explicación diremos que los nodos rotulados como clientes serán aquellos que mediante su navegador y a través de una conexión a Internet o desde la propia red local enviarán las peticiones al servidor, que es el nodo rotulado como máquina servidor y que este a su vez se servirá de los datos almacenados en la base de datos y al que se accederá mediante el bus de datos, ya que el servidor y la base de datos estarán situados en la misma máquina. También es reseñable que nuestro sistema se aprovecha de la propia red para poder realizar conexiones locales, mediante terminales u ordenadores externos a la red local.



3.8 Diseño de la base de datos

3.8.1 Introducción

Para empezar tendremos que definir el concepto de Sistema de gestión de base de datos o DBMS (DataBase Management System). El DBMS consiste en una colección de datos interrelacionados y un conjunto de programas para poder acceder a estos. La colección de datos, normalmente denominada base de datos, contiene información acerca de una actividad determinada. El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

La gestión de datos implica tanto la definición de estructuras para el almacenamiento de información como la provisión de mecanismos para la gestión de esta. Además, los sistemas de bases de datos deben mantener la seguridad de la información almacenada. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos.

3.8.2 Diseño de la estructura conceptual (Modelo Entidad-Relación)

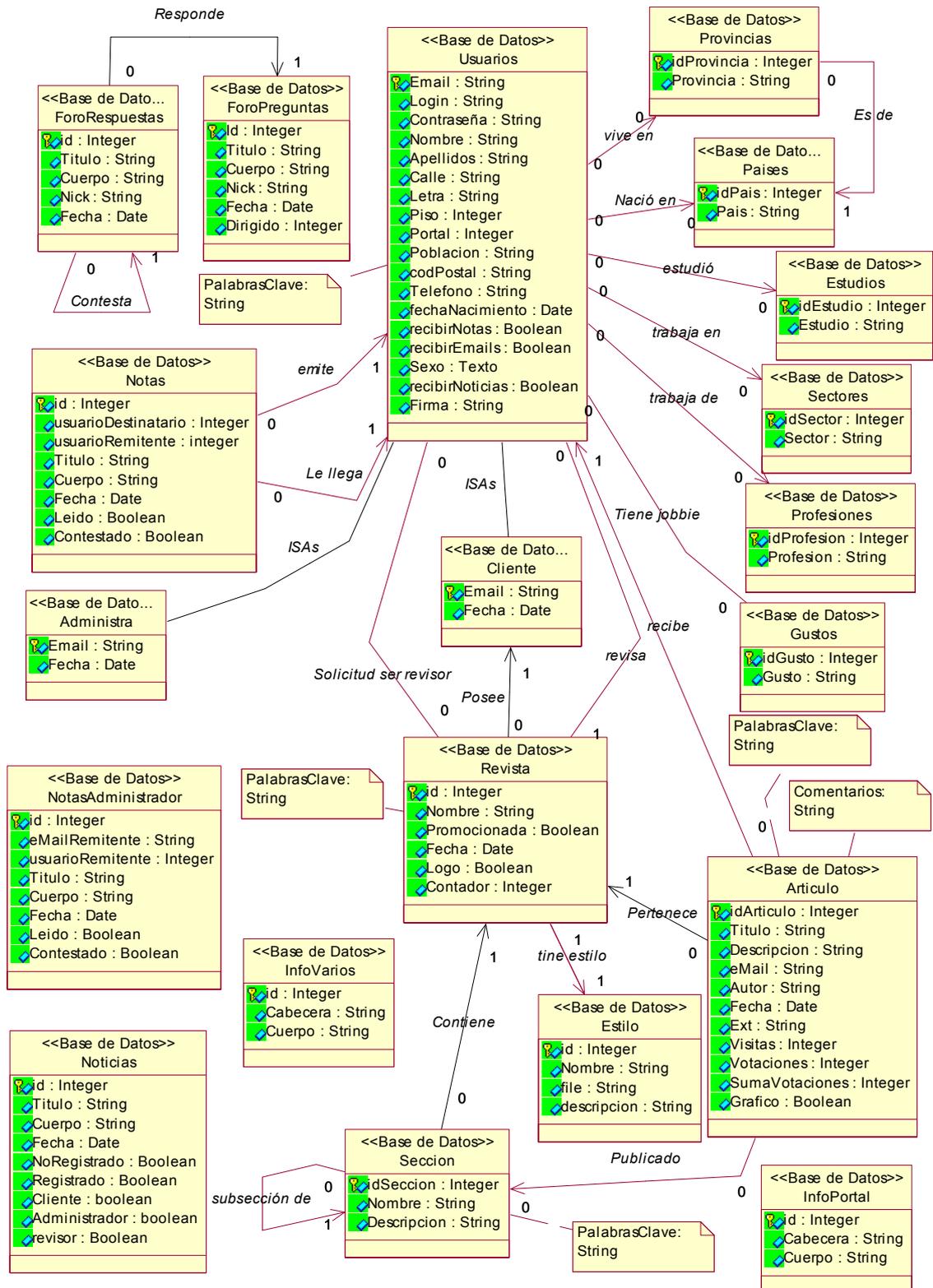
Este modelo representa la percepción del mundo real a través de un esquema gráfico empleando una determinada terminología. En él encontraremos un conjunto de objetos básicos llamados entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver. Cada entidad se distingue de otra por sus características particulares denominadas atributos. Por otro lado están las relaciones que se representan mediante enlaces de unión entre las entidades.

Explicaremos brevemente la simbología empleada en el diagrama siguiente:

- Cada rectángulo representa a las entidades. Su nombre viene especificado en la primera celda del rectángulo.
- En la celda central del rectángulo se especifican los atributos de las entidades. Aquel atributo cuya etiqueta se encuentre precedida por una llave formará parte de la clave primaria de la entidad, los atributos multievaluados se representan mediante una nota.
- Cada línea etiquetada que une dos entidades representa una relación. El contenido de la etiqueta se corresponde con el nombre de la relación en cuestión.
- Las cardinalidad máxima se expresan mediante la propia línea, si se une esta normalmente con la entidad se está expresando una cardinalidad de **n**, en cambio si la unión es mediante una flecha la cardinalidad de la relación respecto a esta es de **1**.
- En cambio la cardinalidad mínima que existen entre dos entidades relacionadas se expresan con una etiqueta situada encima de la línea que las relaciona.

➤ Las relaciones de bitácora se representarán con líneas de color negro.

El modelo Entidad-Relación es el siguiente:



3.8.3 Descripción de las Entidades

Usuario

Esta entidad representa a los distintos usuarios que se pueden registrar en el sistema. Esta entidad cuenta con una serie de atributos en los que se almacenarán los datos necesarios para la identificación y la información relevante sobre ellos.

Atributos			
Nombre	Tipo	Función	Descripción
Email	String	Clave Primaria	Almacenará el Correo electrónico del usuario, que además será utilizado para el proceso de validación
Login	String	Atributo	Nick o login del usuario
Contraseña	String	Atributo	Contraseña del usuario
Nombre	String	Atributo	Nombre del usuario
Apellidos	String	Atributo	Apellidos del usuario
Calle	String	Atributo	Calle en la que vive el usuario
Letra	String	Atributo	Letra del piso en el que vive
Piso	Integer	Atributo	Número del piso en el que vive
Portal	Integer	Atributo	Número del portal
Poblacion	String	Atributo	Población en la que vive
CodPostal	String	Atributo	Código postal de su localidad
Telefono	String	Atributo	El teléfono del usuario
fechaNacimiento	Date	Atributo	La fecha en la que nació
recibirNotas	Boolean	Atributo	Si el usuario quiere recibir las notas mediante E-mail.
recibirEmails	Boolean	Atributo	El usuario permite que otros usuarios les envíen E-mail desde la aplicación
Sexo	String	Atributo	El sexo del usuario
recibirNoticias	Boolean	Atributo	El usuario recibe las noticias en su correo electrónico.
Firma	String	Atributo	Es la firma que se incluyen en los mensajes que envía el usuario.
Palabras Clave	String	Atributo Multievaluado	Palabras que identifica los temas en los que el usuario es un experto, le servirá si este es revisor de alguna revista.

Paises

Representa a los distintos países que hay en el mundo.

Atributos

Nombre	Tipo	Función	Descripción
idPais	Integer	Clave Primaria	Código único de cada país y que le identifica de forma unívoca.
Pais	String	Atributo	El nombre del país.

Provincias

Representa a una serie de provincias.

Atributos

Nombre	Tipo	Función	Descripción
idProvincia	Integer	Clave Primaria	Código único de cada provincia y que la identifica de forma unívoca.
Provincia	String	Atributo	El nombre de la provincia.

Estudios

Representa los distintos estudios que puede tener un usuario.

Atributos

Nombre	Tipo	Función	Descripción
idEstudio	Integer	Clave Primaria	Código único de cada estudio y que lo identifica de forma unívoca.
Estudio	String	Atributo	El nombre del estudio.

Sectores

Representa a los distintos sectores en los que puede estar ejerciendo un usuario su trabajo.

Atributos			
Nombre	Tipo	Función	Descripción
idSector	Integer	Clave Primaria	Código único de cada sector y que lo identifica de forma unívoca.
Sector	String	Atributo	El nombre del sector.

Profesiones

Representa las distintas profesiones que un usuario puede ejercer.

Atributos			
Nombre	Tipo	Función	Descripción
idProfesion	Integer	Clave Primaria	Código único de cada profesión y que lo identifica de forma unívoca.
Profesion	String	Atributo	El nombre de la profesión.

Gustos

Representa los distintos hobbies o gustos que pueda tener un usuario.

Atributos			
Nombre	Tipo	Función	Descripción
idGusto	Integer	Clave Primaria	Código único de cada gusto y que lo identifica de forma unívoca.
Gusto	String	Atributo	El nombre del gusto.

Cliente

Representa a los distintos usuarios que son también clientes del portal y por tanto pueden crear revistas.

Atributos			
Nombre	Tipo	Función	Descripción
Email	Integer	Clave Primaria	Almacenará el Correo electrónico del cliente, que además será utilizado para el proceso de validación
Fecha	Date	Atributo	La fecha en la que el usuario paso a ser también un cliente.

Administra

Representa a los distintos usuarios que son también administradores del portal.

Atributos			
Nombre	Tipo	Función	Descripción
Email	Integer	Clave Primaria	Almacenará el Correo electrónico del administrador, que además será utilizado para el proceso de validación
Fecha	Date	Atributo	La fecha en la que usuario paso a ser también un Administrador.

ForoPreguntas

Esta entidad representa todas las preguntas que se han podido realizar en el foro.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada pregunta del foro y que la identifica de forma unívoca.
Titulo	String	Atributo	Es el título que se la puesto a la pregunta.
Cuerpo	String	Atributo	Es la pregunta en sí.
Nick	String	Atributo	Es el nick del usuario que realizó la pregunta.
Fecha	Date	Atributo	Fecha en la que se realizó la pregunta
Dirigido	Integer	Atributo	Nos indica a que tipo de usuario va dirigida la pregunta.

ForoRespuestas

Esta entidad representa todas las respuestas que se han podido realizar en el foro.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada respuesta del foro y que la identifica de forma unívoca.
Titulo	String	Atributo	Es el título que se la puesto a la respuesta.
Cuerpo	String	Atributo	Es la respuesta en sí.
Nick	String	Atributo	Es el nick del usuario que realizó la respuesta.
Fecha	Date	Atributo	Fecha en la que se realizó la respuesta

Notas

Esta entidad representa todas las notas que se envían unos usuarios a otros.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada nota y que la identifica de forma unívoca.
Usuario Destinatario	Integer	Atributo	Es el tipo de usuario al que va dirigida. Recordemos que un usuario puede tener varios roles: Administrador, cliente, etc.
Usuario Remitente	Integer	Atributo	Es el tipo de usuario que ha enviado la nota.
Título	String	Atributo	Es el título de la nota enviada.
Cuerpo	String	Atributo	Es la nota en sí.
Fecha	Date	Atributo	Fecha en la que se envió la nota.
Leído	Boolean	Atributo	Nos indica si el usuario al que va dirigida la nota la ha leído.
Contestado	Boolean	Atributo	Nos indica si el usuario al que va dirigida la nota la ha contestado.

NotasAdministrador

Se podrán enviar notas a cualquier administrador sin importarnos cual de ellos la lee, esta entidad almacena los datos de estas notas.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada nota y que la identifica de forma unívoca.
Email Remitente	String	Atributo	Es el E-mail de quien envía la nota.
Usuario Remitente	Integer	Atributo	El tipo de usuario que envía la nota, pudiendo ser alguien que no esté registrado.
Título	String	Atributo	Es el título de la nota.
Cuerpo	String	Atributo	Es la nota en sí.
Fecha	Date	Atributo	Es la fecha en la que se envió la nota.
Leído	Boolean	Atributo	Si algún administrador ha leído la nota.
Contestado	Boolean	Atributo	Si algún administrador ha contestado la nota.

Revista

Almacena los datos referentes a cada una de las revistas que irán creando los distintos clientes.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada revista y que la identifica de forma unívoca.
Nombre	String	Atributo	Es el nombre de la revista.
Promocionada	Boolean	Atributo	Si esta revista y sus artículos aparecen en la búsqueda de revistas y artículos.
Fecha	Date	Atributo	Es la fecha en la que se creó la revista.
Logo	Boolean	Atributo	Si tiene la revista un logotipo asociado.
Contador	Integer	Atributo	Nos indica las visitas que ha tenido la revista.
PalabrasClave	String	Atributo multievaluado	Palabras que identifican los temas que trata la revista.

InfoPortal

Almacena información de ayuda relativa al portal y que se mostrará en la aplicación como un texto informativo.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de elemento informativo de la revista y que la identifica de forma unívoca.
Cabecera	String	Atributo	Es la cabecera del texto informativo de la revista.
Cuerpo	String	Atributo	Es el texto informativo en sí.

InfoVarios

Al igual que infoPortal almacena información de ayuda relativa al portal y que se mostrará en la aplicación como un texto informativo, pero en este caso será relativo a eventos provocados como son la realización correcta de operaciones, errores, etc. Se ha decido crear otra tabla pues podría ser interesante el realizar un control sobre los errores u operaciones realizadas con éxito y aquí con una pequeña modificación de la tabla se podrían realizar su control.

Es decir esta tabla se creo para dejar el diseño más abierto y permitir en un futuro el diseño e implementación de control y monitorización de errores provocados por la aplicación.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de elemento informativo de la revista y que la identifica de forma unívoca.
Cabecera	String	Atributo	Es la cabecera del texto informativo de la revista.
Cuerpo	String	Atributo	Es el texto informativo en sí.

Articulo

Almacena información sobre los artículos que se publican en las distintas revistas.

Atributos			
Nombre	Tipo	Función	Descripción
idArticulo	Integer	Clave Primaria	Código único de la revista y que la identifica de forma unívoca.
Titulo	String	Atributo	Es el título del artículo.
Descripcion	String	Atributo	Es una breve descripción sobre el artículo.
Email	String	Atributo	Es el Email del autor del artículo
Autor	String	Atributo	Es el nombre del autor del artículo
Fecha	String	Atributo	Fecha en la que se envía el artículo
Ext	String	Atributo	Nos indica si el artículo está contenido en un documento .doc ó .pdf.
Visitas	Integer	Atributo	Nos indica cuantas personas han visto el artículo.
Votaciones	Integer	Atributo	Número de votaciones que ha tenido el artículo.
Suma Votaciones	Integer	Atributo	Es la suma de todas las votaciones que ha tenido el artículo.
Grafico	Boolean	Atributo	Nos indica si el artículo tiene asociado o no una imagen.
Palabras Clave	String	Atributo multievaluado	Palabras que identifican los temas que trata el artículo.

Noticias

Almacena las noticias que se publican en el portal.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada noticia y que la identifica de forma unívoca.
Titulo	String	Atributo	Título de la noticia.
Cuerpo	String	Atributo	Es la noticia en sí.
Fecha	Date	Atributo	Es la fecha en la cual se ha publicado el artículo.
NoRegistrado	Boolean	Atributo	Si esta noticia está dirigida a los usuarios no registrados.
Registrado	Boolean	Atributo	Si esta noticia está dirigida a los usuarios registrados.
Cliente	Boolean	Atributo	Si la noticia está dirigida a los clientes.
Administrador	Boolean	Atributo	Si la noticia está dirigida a los administradores.
Revisor	Boolean	Atributo	Si la noticia está dirigida a los revisores.

Seccion

Almacena información sobre las secciones en las que están divididas las distintas revistas.

Atributos			
Nombre	Tipo	Función	Descripción
idSeccion	Integer	Clave Primaria	Código único de cada sección que la identifica de forma unívoca.
Nombre	String	Atributo	Nombre de la sección.
Descripcion	String	Atributo	Es una breve descripción sobre el contenido que albergará la sección.
PalabrasClave	String	Atributo multievaluado	Palabras que identifican los temas que trata la sección.

Estilo

Almacena los distintos estilos que podrá tener una revista.

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código único de cada estilo que le identifica de forma unívoca.
Nombre	String	Atributo	Nombre del estilo
Descripcion	String	Atributo	Es una breve descripción sobre la apariencia que tendría la revista con ese estilo.

3.8.4 Descripción de las relaciones entre entidades

Responde

Nos relaciona las respuestas del foro con las preguntas a las cuales está respondiendo.

Entidades que relaciona	ForoRespuestas		ForoPreguntas	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	1	1

Atributos			
Nombre	Tipo	Función	Descripción
Id (Foro Respuesta)	Integer	Clave Primaria	Código de la respuesta
Id (Foro Pregunta)	Integer	Clave Primaria	Código de la pregunta

Contesta

Relaciona las respuestas que contestan a otras respuestas.

Entidades que relaciona	ForoRespuestas		ForoRespuestas	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	1	1

Atributos			
Nombre	Tipo	Función	Descripción
Id (Foro Respuesta)	Integer	Clave Primaria	Código de la respuesta que es contestación
Id (Foro Respuesta)	Integer	Clave Primaria	Código de la respuesta a la cual contesta

Vive en

Nos relaciona al usuario con la provincia en la cual vive.

Entidades que relaciona	Usuario		Provincias	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	1

Atributos			
Nombre	Tipo	Función	Descripción
idProvincia	Integer	Clave Primaria	Código de la provincia
Email	String	Clave Primaria	Correo del usuario

Es de

Nos relaciona las provincias con el país donde se encuentra.

Entidades que relaciona	Provincias		Países	
Cardinalidad	mínima	máxima	mínima	máxima
	0	n	1	1

Atributos

Nombre	Tipo	Función	Descripción
idProvincia	Integer	Clave Primaria	Código de la provincia
idPais	Integer	Clave Primaria	Código del país

Nació en

Esta relación representa el país en el que ha nacido el usuario.

Entidades que relaciona	Usuario		Países	
Cardinalidad	mínima	máxima	mínima	máxima
	0	n	0	1

Atributos

Nombre	Tipo	Función	Descripción
idPais	Integer	Clave Primaria	Código del país
Email	String	Clave Primaria	Email del usuario

Estudió

Esta relación representa los estudios que tiene el usuario.

Entidades que relaciona	Usuario		Estudios	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	1

Atributos			
Nombre	Tipo	Función	Descripción
idEstudio	Integer	Clave Primaria	Código del estudio
Email	String	Clave Primaria	Email del usuario

Trabaja en

Nos relaciona al usuario con el sector en el que ejerce su profesión.

Entidades que relaciona	Usuario		Sectores	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	1

Atributos			
Nombre	Tipo	Función	Descripción
idSector	Integer	Clave Primaria	Código del sector
Email	String	Clave Primaria	Email del usuario

Trabaja de

En este caso nos relaciona el usuario con la profesión que tiene.

Entidades que relaciona	Usuario		Profesiones	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	1

Atributos			
Nombre	Tipo	Función	Descripción
idProfesion	Integer	Clave Primaria	Código de la profesión
Email	String	Clave Primaria	Email del usuario

Tiene jobbie

Esta relación representa los jobbies o gustos que tiene el usuario.

Entidades que relaciona	Usuario		Gustos	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	n

Atributos			
Nombre	Tipo	Función	Descripción
idGusto	Integer	Clave Primaria	Código del gusto
Email	String	Clave Primaria	Email del usuario

Recibe

Esta relación representa los artículos que esta revisando un usuario para ser publicado en una revista (en la cual el usuario es revisor).

Entidades que relaciona	Usuario		Articulo	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	0	n

Atributos

Nombre	Tipo	Función	Descripción
idArticulo	Integer	Clave Primaria	Código del artículo
Email	String	Clave Primaria	Email del usuario

Emite

Nos identifica al usuario que ha enviado una nota a otro usuario.

Entidades que relaciona	Usuario		Notas	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	0	n

Atributos

Nombre	Tipo	Función	Descripción
id	Integer	Clave Primaria	Código de la nota
Email	String	Clave Primaria	Email del usuario

Le llega

En este caso se identifica al usuario que recibe una nota de otro.

Entidades que relaciona	Usuario		Notas	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	0	n

Atributos			
Nombre	Tipo	Función	Descripción
id	Integer	Clave Primaria	Código de la nota
Email	String	Clave Primaria	Email del usuario

Solicitud ser revisor

Esta relación identifica las solicitudes que pueden hacer los usuarios para ser revisores en una determinada revista.

Entidades que relaciona	Revista		Usuario	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	n

Atributos			
Nombre	Tipo	Función	Descripción
Id	Integer	Clave Primaria	Código de la revista
Email	String	Clave Primaria	Email del usuario
Exposición	String	Atributo	La exposición de la solicitud para ser revisor
Fecha	Date	Atributo	Fecha en la que se ha realizado la solicitud

Posee

Relaciona las revistas con el cliente que la ha creado.

Entidades que relaciona	Cliente		Revista	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	0	n

Atributos			
Nombre	Tipo	Función	Descripción
Email	String	Clave Primaria	Email del cliente
Id	Integer	Clave Primaria	Código de la revista

Contiene

Esta relación nos identifica las secciones con las revistas en las cual están contenidas.

Entidades que relaciona	Seccion		Revista	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	1	1

Atributos			
Nombre	Tipo	Función	Descripción
idSección	Int	Clave Primaria	Código de la sección
id	Int	Clave Primaria	Código de la revista

Subsección de

Nos identifica a las secciones que están contenidas dentro de otras secciones.

Entidades que relaciona	Seccion		Seccion	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	1	n

Atributos			
Nombre	Tipo	Función	Descripción
idSeccion	Int	Clave Primaria	Código de la sección
idSeccion	Int	Clave Primaria	Código de la sección

Publicado

Relaciona el artículo con la sección en la cual es publicado.

Entidades que relaciona	Seccion		Articulo	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	n

Atributos			
Nombre	Tipo	Función	Descripción
idSeccion	Int	Clave Primaria	Código de la sección
idArticulo	Int	Clave Primaria	Código del artículo
Fecha	Date	Atributo	Fecha de la publicación

Tiene estilo

Nos identifica la relación existente entre la revista y el estilo que tiene.

Entidades que relaciona	Estilo		Revista	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	1	n

Atributos

Nombre	Tipo	Función	Descripción
Id (estilo)	Integer	Clave Primaria	Código del estilo
Id (Revista)	Integer	Clave Primaria	Código de la revista

Pertenece

Esta relación nos identifica la revista en la que pertenece un artículo.

Entidades que relaciona	Revista		Artículo	
	mínima	máxima	mínima	máxima
Cardinalidad	1	1	0	n

Atributos

Nombre	Tipo	Función	Descripción
id	Integer	Clave Primaria	Código de la revista
idArticulo	Integer	Clave Primaria	Código del artículo

Revisa

Esta relación nos identifica a un usuario con la revista que revisa.

Entidades que relaciona	Usuario		Revista	
	mínima	máxima	mínima	máxima
Cardinalidad	0	n	0	n

Atributos			
Nombre	Tipo	Función	Descripción
ID	Integer	Clave Primaria	Código de la revista
Email	String	Clave Primaria	Email del usuario
Fecha	Date	Atributo	Fecha en la que el usuario se ha hecho revisor de la revista

3.8.5 Diseño lógico (Paso a tablas y normalización)

En primer lugar empezaremos pasando a tablas las relaciones, ya que habrá alguna que no generará tabla y por tanto alguna de las entidades que relaciona pasará a tener sus atributos.

Responde

Es una relación de bitácora por eso no genera tabla y es absorbida por la tabla de “muchos”, en este caso ForoRespuestas.

Contestada

Al igual que en el caso anterior, la relación desaparece por ser uno a muchos y es absorbida por la entidad “muchos” que es ForoRespuestas.

Vive en

Esta también desaparece ya que le sucede lo mismo que a las anteriores, en este caso es absorbida por Usuario.

Es de

Al igual que el resto, esta es absorbida por provincias.

Nació en

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Usuario.

Estudió

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Usuario.

Trabaja en

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Usuario.

Trabaja de

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Usuario.

Tiene Jobbie (a partir de ahora Jobbies)

<i>Atributos de Jobbies</i>	Jobbie (idGusto)	Email
-----------------------------	-------------------------	--------------

Jobbies	
Clave primaria	Email \cup Jobbie
Claves candidato	Email \cup Jobbie
Claves Externa	Email (de Crítico), Jobbie (IdGusto de Gustos)

Recibe

<i>Atributos de Recibe</i>	idArticulo	Email
----------------------------	------------	-------

Recibe	
Clave primaria	idArticulo \cup Email
Claves candidato	idArticulo \cup Email
Claves Externa	idArticulo (de articulo) \cup Email (de Usuario)
Reglas de borrado	Si el usuario es eliminado se eliminarán las tuplas de esta tabla relacionadas con ese usuario ya que dejará de revisarlo y le dará la oportunidad a otros usuario para que tenga nuevo revisor.

Emite

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Notas.

Le llega

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Notas.

Solicitud ser revisor a partir de ahora (SolicitudSerServidor)

<i>Atributos SolicitudSerServidor</i>	IdRevi	Email	Exposicion	Fecha
---------------------------------------	--------	-------	------------	-------

SolicitudSerRevisor	
Clave primaria	idRevi \cup Email
Claves candidato	idRevi \cup Email
Claves Externa	idRevi (Id de Revista), Email (de Usuario)
Dependencias	idRevi,Email \rightarrow Exposicion ,Fecha
Reglas de borrado	Se eliminará una solicitud si se elimina la revista o el usuario.

Posee

Es una relación de bitácora por eso no genera tabla y es absorbida por la tabla de “muchos”, en este caso es Revista.

Contiene

Es una relación de bitácora por eso no genera tabla y es absorbida por la tabla de “muchos”, en este caso Seccion.

Subsección de

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Seccion.

Pertenece

Es una relación de bitácora por eso no genera tabla y es absorbida por la tabla de “muchos”, en este caso Seccion.

Tiene estilo

Desaparece esta relación por ser una relación uno a muchos y es absorbida por la entidad “muchos” que relaciona, en este caso es Revista.

Publicado

<i>Atributos Publicado</i>	idSeccion	idArticulo	Fecha
----------------------------	-----------	------------	-------

Publicado	
Clave primaria	idSeccion \cup idArticulo
Claves candidato	idSeccion \cup idArticulo
Claves Externa	idSeccion (de Seccion), idArticulo (de Articulo)
Dependencias	disección,idArticulo \rightarrow Fecha
Reglas de borrado	Si se elimina un artículo o una sección, se deberán de eliminar las tuplas que estén relacionadas con el artículo o la sección.

Revisa

<i>Atributos Revisa</i>	Email	Id	Fecha
--------------------------------	--------------	-----------	--------------

Revisa	
Clave primaria	Email ∪ Id
Claves candidato	Email ∪ Id
Claves Externa	Email (de usuario), Id (de revista)
Dependencias	Email, Id → Fecha
Reglas de borrado	Si se elimina alguna revista o usuario, se deberán eliminar las tuplas de esta tabla que estén relacionadas.

Usuario

<i>Atributos Usuario</i>	eMail	Login	Contraseña	Nombre	Apellidos	Calle
	Letra	Piso	Portal	Población	CódPostal	Provincia
	Teléfono Contacto	Fecha Nacimiento	Estudios	Sector	Ocupación	Recibir Notas
	Recibir Emails	Sexo	Pais	Recibir Noticias	Firma	

Usuario	
Clave primaria	Email
Claves candidato	Email y Login
Calves Externas	Provincia (idProvincia de Probincias), Pais (idPais de Paises), Estudios (idEstudio de Estudios), Sector (idSector de Sectores) y Ocupación (idProfesion de Profesiones)
Dependencias	Email → Login, Contraseña, Nombre, Apellidos, calle, letra, piso, portal, población, CodPosta, Provincia, TelefonoContacto, FechaNacimiento, Estudios, Sector, Ocupación, RecibirNotas, RecibirEmails, sexo, Pais, RecibirNoticias, Fima. Login → Email, Contraseña, Nombre, Apellidos, calle, letra, piso, portal, población, CodPosta, Provincia, TelefonoContacto, FechaNacimiento, Estudios, Sector, Ocupación, RecibirNotas, RecibirEmails, sexo, Pais, RecibirNoticias, Fima.
Reglas de borrado	Cuando se borre a un usuario, se deberán los usuarios generalizados (ISAs) de Cliente y Administra.

Paises

<i>Atributos Paises</i>	idPais	Pais
--------------------------------	---------------	-------------

Paises	
Clave primaria	idPais
Claves candidato	idPais
Dependencias	idPais → Pais

Provincias

<i>Atributos Provincias</i>	idProvincia	Provincia	idPais
------------------------------------	--------------------	------------------	---------------

Provincias	
Clave primaria	idProvincia
Claves candidato	idProvincia
Claves Externa	idPais (de Paises)
Dependencias	idProvincia → Provincia, idPais
Reglas de borrado	Si se elimina un pais, se eliminarán todas las provincias que dependan de este

Estudios

<i>Atributos estudios</i>	idEstudio	Estudio
----------------------------------	------------------	----------------

Estudios	
Clave primaria	idEstudio
Claves candidato	idEstudio
Dependencias	idEstudio → Estudio

Sectores

<i>Atributos Sectores</i>	idSector	Sector
----------------------------------	-----------------	---------------

Sectores	
Clave primaria	idSector
Claves candidato	idSector
Dependencias	idSector → Sector

Profesiones

<i>Atributos Profesiones</i>	idProfesion	Profesion
-------------------------------------	--------------------	------------------

Relación - entidad	
Clave primaria	idProfesion
Claves candidato	idProfesion
Dependencias	idProfesion → Profesion

Gustos

<i>Atributos Gustos</i>	idGusto	Gusto
--------------------------------	----------------	--------------

Gustos	
Clave primaria	idGusto
Claves candidato	idGusto
Dependencias	idGusto → Gusto

Cliente

<i>Atributos Cliente</i>	Email	Fecha
---------------------------------	--------------	--------------

Cliente	
Clave primaria	Email
Claves candidato	Email
Claves Externa	Email (de Usuario)
Dependencias	Email → Fecha
Reglas de borrado	Se eliminan todos los clientes, si se elimina el usuario asociado.

Administra

<i>Atributos Administra</i>	Email	Fecha
------------------------------------	--------------	--------------

Administra	
Clave primaria	Email
Claves candidato	Email
Claves Externa	Email (de Usuario)
Dependencias	Email → Fecha
Reglas de borrado	Se eliminan todos los administradores, si se elimina el usuario asociado.

foroPreguntas

<i>Atributos foroPreguntas</i>	ID	Titulo	Cuerpo	Nick	Fecha	Dirigido
---------------------------------------	-----------	---------------	---------------	-------------	--------------	-----------------

foroPreguntas	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → Titulo, cuerpo, Nick, Fecha, Dirigido
Reglas de borrado	Si se borra alguna pregunta se borrarán las tuplas de ForoRespuestas que estén relacionadas.

foroRespuestas

<i>foroRespuestas</i>	ID	IDPregunta	IDRespuesta	Titulo	Cuerpo	Nick
	Fecha					

Relación - entidad

Clave primaria	ID
Claves candidato	ID
Claves Externa	IDPregunta (ID de ForoPregunta), IDRespuesta (ID de otra tupla de ForoRespuesta)
Dependencias	ID → IDPregunta, IDRespuesta, Titulo, Cuerpo, Nick, Fecha
Reglas de inserción	Si se inserta una respuesta a una respuesta, el idRespuesta debe corresponder con un ID existente en idRespuestas, en caso contrario su valor debe de ser 0.
Reglas de borrado	Si se elimina una pregunta se eliminarán todas las respuestas que dependan de ellas, de igual manera si se elimina una respuesta, se eliminarán las respuestas que dependan de esta.

Notas

<i>Atributos Notas</i>	ID	eMail Destinatario	eMail Remitente	Usuario Destinatario	Usuario Remitente	Titulo
	Cuerpo	Fecha	Leido	Contestado		

Notas

Clave primaria	ID
Claves candidato	ID
Claves Externa	EmailDestinatario (Email de Esuario), EmailRemitente (Email de Usuario)
Dependencias	ID → EmailDestinatario, EmailRemitente, UsuarioDestinatario, UsuarioRemitente, Titulo, Cuerpo, Fecha, Leido, Contestado.
Reglas de borrado	Si desaparece un usuario, se eliminarán las notas que dependan de él.

NotasAdministradores

<i>Atributos NotasAdministradores</i>	ID	Email Remitente	Usuario Remitente	Titulo	Cuerpo	Fecha
	Leido	Contestado				

NotasAdministradores	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → EmailRemitente, UsuarioRemitente, Titulo, Cuerpo, Fecha, Leido, Contestado

Revista

<i>Atributos Revista</i>	ID	Nombre	Promocionada	Descripcion	eMail	Fecha
	Logo	Estilo	Contador			

Revista	
Clave primaria	ID
Claves candidato	ID
Claves Externa	Email (de usuario), Estilo (ID de Estilo)
Dependencias	Id → Nombre, Promocionada, Descripción, Email, Fecha, Logo, Estilo, Contador
Reglas de borrado	Si se elimina a un usuario se eliminará las revistas relacionadas con él. Si se elimina la revista, se eliminan los artículos y secciones que dependan de esta.

InfoPortal

<i>Atributos InfoPortal</i>	ID	Cabecera	Cuerpo
-----------------------------	----	----------	--------

InfoPortal	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → Cabecera, Cuerpo

InfoVarios

InfoVarios tienen los mismos campos que InfoPortal, no unimos las dos tablas para permitir dejar el diseño más abierto. Esta tabla muestra información excepcional, en un futuro quizás se quiera saber cuantas veces se producen estas situaciones o a que usuario se les produce. No obstante en el proyecto actual no se va a realizar estos registros pero dejamos el diseño abierto para que se pueda realizar en un futuro una modificación de esta tabla para que registre estas situaciones excepcionales.

<i>Atributos InfoVarios</i>	ID	Cabecera	Cuerpo
------------------------------------	-----------	-----------------	---------------

InfoVarios	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → Cabecera, Cuerpo

Articulo

<i>Atributos Articulo</i>	IDArticulo	ID	Titulo	Descripcion	Email	Autor
	Fecha	Ext	Visitas	Votaciones	SumaVotaciones	Grafico

Articulo	
Clave primaria	IDArticulo
Claves candidato	IDArticulo
Claves Externa	ID (ID de revista)
Dependencias	IDArticulo → ID, Titulo, Descripción, Email, Autor, Fecha, Ext, Visitas, Votaciones, SumaVotaciones, Grafico.
Reglas de borrado	Si se elimina la revista, se eliminarán los artículos asociados con ella.

Noticias

<i>Atributos Noticias</i>	ID	Título	Cuerpo	Fecha	NoRegistrado	Registrado
	Cliente	Administrador	Revisor			

Noticias	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → Título, Cuerpo, Fecha, NoRegistrado, Registrado, Cliente, Administrador, Revisor

Seccion

<i>Atributo Seccion</i>	IDSeccion	ID	Nombre	Descripcion	Padre
-------------------------	-----------	----	--------	-------------	-------

Seccion	
Clave primaria	IDsección
Claves candidato	IdSección
Claves Externa	ID (de Revista)
Dependencias	Disección → Id, Nombre, Descripción, Padre

Estilo

<i>Atributo Estilo</i>	ID	Nombre	File	Descripción
------------------------	----	--------	------	-------------

Estilo	
Clave primaria	ID
Claves candidato	ID
Dependencias	ID → Nombre, File, Descripción

comentariosArticulos

Aparece de un atributo multievaluado de la entidad Articulo.

<i>Atributos ComentariosArticulos</i>	IdArticulo	Comentario	Fecha	Autor
---	-------------------	-------------------	--------------	--------------

comentariosArticulos	
Clave primaria	idArticulo, Comentario
Claves candidato	idArticulo, Comentario
Claves Externa	idArticulo (de Articulo)
Dependencias	idArticulo, Comentario → Fecha, Autor
Reglas de borrado	Si se elimina un artículo, se eliminarán los comentarios asociados.

PalabrasClaveArticulo

Aparece de un atributo multievaluado de la entidad Articulo.

<i>Atributos PalabrasClaveArticulo</i>	idArticulo	Palabra
--	-------------------	----------------

palabrasClaveArticulo	
Clave primaria	idArticulo, Palabra
Claves candidato	idArticulo, Palabra
Claves Externa	idArticulo (de Articulo)
Reglas de borrado	Si se elimina un artículo, se eliminarán las palabras clave asociadas.

PalabrasClaveRevisor

Aparece de un atributo multievaluado de la entidad Usuario.

<i>Atributos PalabrasClaveRevisor</i>	Email	Palabra
---------------------------------------	--------------	----------------

PalabrasClaveRevisor	
Clave primaria	Email, Palabra
Claves candidato	Email, Palabra
Claves Externa	Email (de usuario)
Reglas de borrado	Si se elimina al usuario, se eliminarán sus palabras clave.

PalabrasClaveRevista

Aparece de un atributo multievaluado de la entidad Revista.

<i>Atributos PalabrasClaveRevista</i>	ID	Palabra
---------------------------------------	-----------	----------------

PalabrasClaveRevista	
Clave primaria	ID, Palabra
Claves candidato	ID, Palabra
Claves Externa	ID (ID de revista)
Reglas de borrado	Si se elimina una revista, se eliminarán las palabras clave asociadas con esta.

PalabrasClaveSeccion

Aparece de un atributo multievaluado de la entidad Seccion.

<i>Atributos PalabrasClaveSeccion</i>	ID	Palabra
---------------------------------------	-----------	----------------

PalabrasClaveSeccion	
Clave primaria	ID, Palabra
Claves candidato	ID, Palabra
Reglas de borrado	Si se elimina una sección se eliminarás las palabras clave asociadas con esta.

3.9 Bibliografía

[1] Análisis y diseño orientado a Objetos
Cuaderno N°1, 2ª Edición (Abril de 2000)
Juan Manuel Cueva Lovelle
Universidad de Oviedo

[2] JavaServer Pages – Manual de usuario y tutorial
Agustín Froufe
Ra-Ma

[3] Patrones de diseño y arquitectónicos
Aquilino A. Juan
Universidad de Oviedo

[4] página de la Racional Rose
<http://www.rational.com>

[5] Especificación de UML
<http://www.rational.com/media/uml/post.pdf>

[6] Fundamentos de Bases de Datos – Tercera edición
Abraham Silberschatz, Henry F. Korth y S. Sudarshan
Mc Graw Hill

[7] Tutorial de UML
Departamento de Ciencias de la Computación. Universidad de Chile
<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>