

3 Diseño e Implementación

En la elaboración de este proyecto se han utilizado diferentes herramientas informáticas, dependiendo de la etapa de diseño o implementación.

Se ha utilizado la herramienta StarUML para la especificación y descripción de los casos de uso y diagramas de secuencia de la aplicación.

Los diagramas entidad – relación y grafo relacional de la base de datos se han creado por medio de la herramienta Studio CASE V1.0.

Para poder crear la base de datos del sistema, y poder realizar las pruebas de la aplicación, se ha instalado la herramienta Wamp. Esta aplicación permite instalar en un solo paquete un servidor Apache, MySQL y PHP.

La programación de las distintas partes, ya sea PHP, HTML, CSS o Javascript se ha realizado sobre el editor Komodo Project File.

3.1 Diseño del Sistema de Información (UML). Herramienta StarUML

Para el desarrollo de los distintos casos de uso y diagramas de secuencia de la aplicación en Lenguaje Unificado de Modelado, se ha utilizado la herramienta *StarUML*, de la que a continuación se muestra un ejemplo.

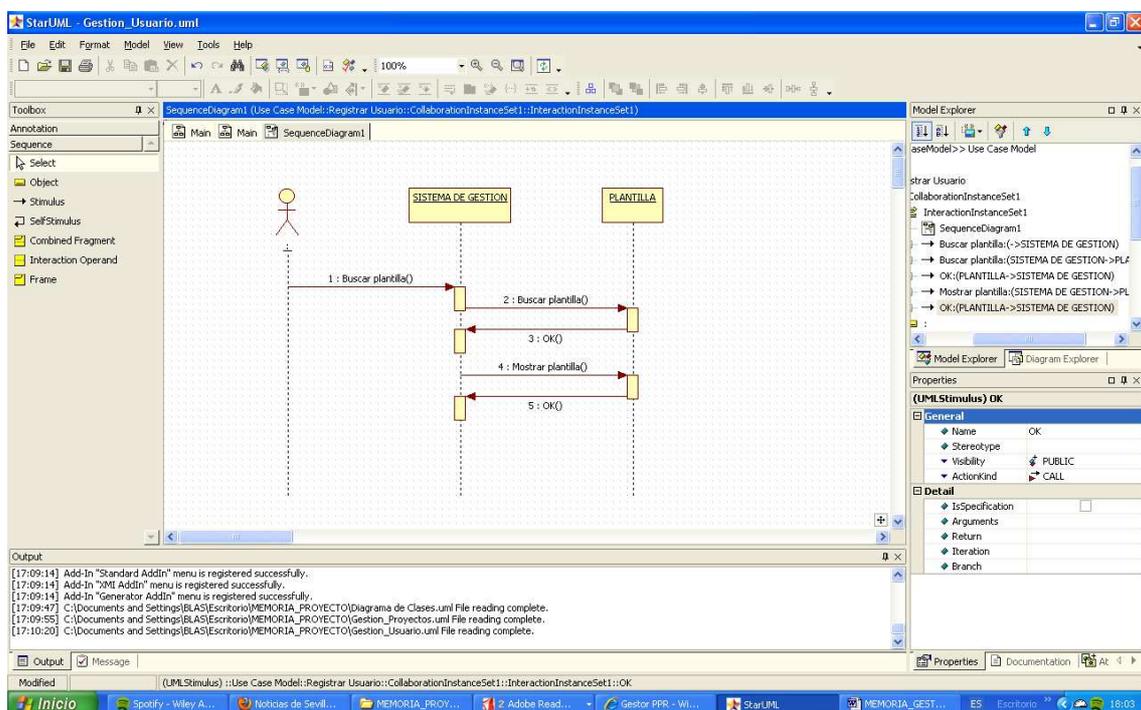


Figura 3.1.1 1
Pantalla de ejemplo de StarUML

3.2 Herramienta Studio CASE v1.0

Para la creación de los diagramas de Entidad - Relación de la base de datos se ha usado la herramienta Studio Case, que permite obtener el diagrama completo de una manera muy rápida y sencilla.

Un ejemplo de esta aplicación es la siguiente captura de pantalla:

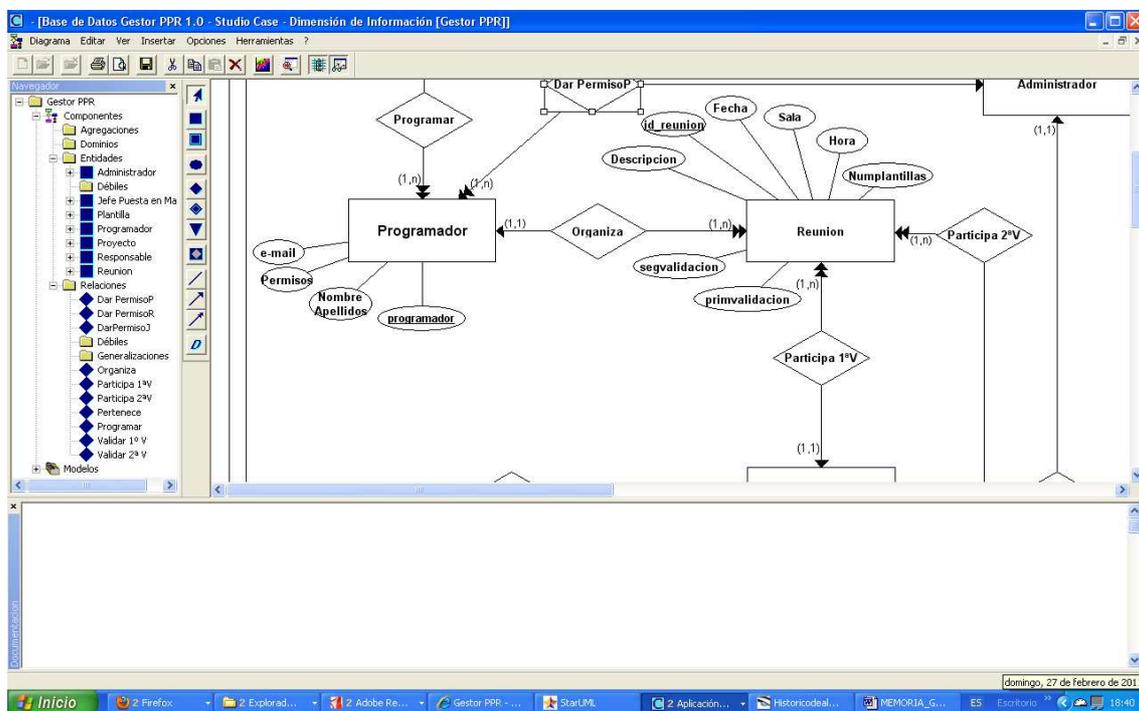


Figura 3.1.1 2

Pantalla de ejemplo de Studio Case

3.3 Editor de Texto Komodo Edit 5

La programación de los distintos archivos en *PHP*, *HTML*, *CSS* y *Javascript* se ha realizado con el editor *Komodo Project File* que facilita la creación y modificación de dichos archivos debido a las opciones ofrecidas.

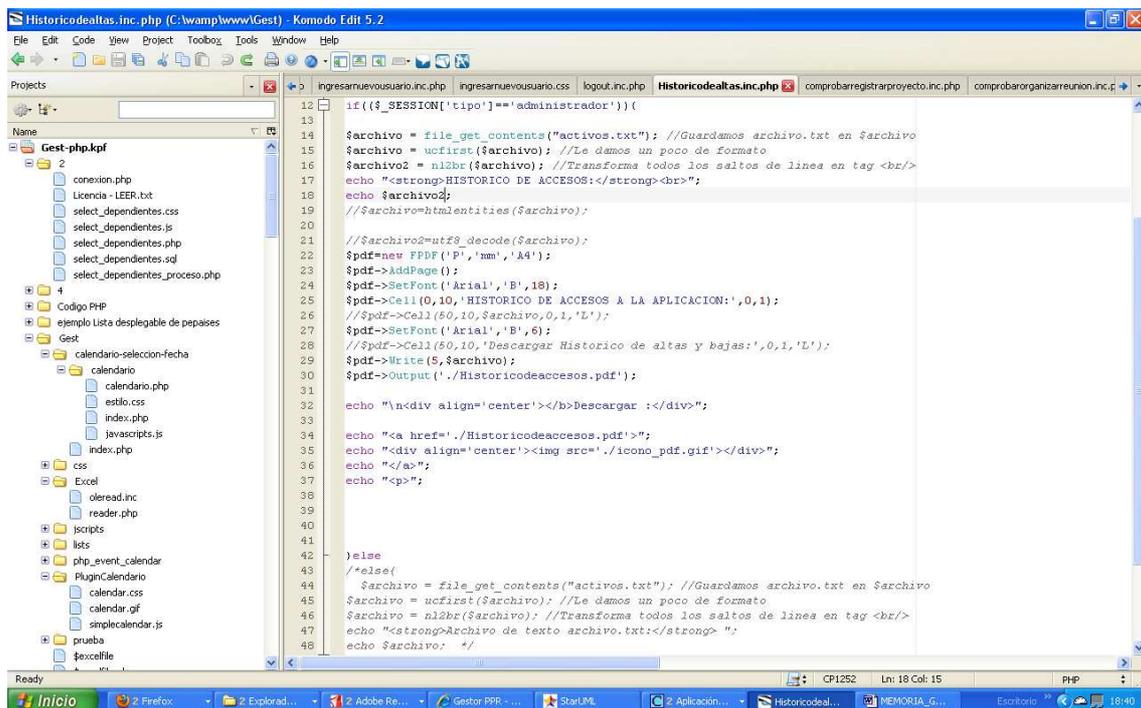


Figura 3.1.3 1

Captura de Editor Komodo

3.4 Herramienta *WAMP*

La aplicación a desarrollar en el proyecto necesita de una serie de herramientas para comunicarse con la base de datos. Se necesita también crear una base de datos en modo local para hacer pruebas.

Es necesario instalar y configurar *MySQL*, *PHPmyAdmin* y el servidor *Apache* para poder crear el servidor web que atienda las peticiones a la base de datos.

Pero para facilitar la complicada tarea de instalar y configurar correctamente todas estas herramientas, existen una serie de paquetes ya elaborados y preparados para instalar y configurar de manera automática todas las herramientas necesarias. Esto es, el paquete *WampServer*.

WampServer es por tanto un sistema indicado para los usuarios que no tienen instalado en el sistema ninguna de los programas necesarios para programar en PHP (*Apache*, *PHP* y *MySQL*) ya que realiza una instalación completa y desde cero.

También pueden utilizar este programa los usuarios que disponen de *Apache*, *PHP* y/o *MySQL* en su sistema.

✓ **Aplicaciones contenidas en la herramienta *WAMP***

El software contiene los siguientes programas:

- Apache 2.2.11
- PHP 5.3.0
- MySQL. Base de datos más extendida para utilizar con PHP.
- PHPMyAdmin. Herramienta que permite la administración con MySQL, pensada para ser una interfaz web, fue desarrollada con código PHP.

3.5 Visión global de la aplicación

3.5.1 Descripción textual de las necesidades de la aplicación

En este punto se pretende dar una visión general de la estructura y el funcionamiento del Gestor PPR. El objetivo fundamental de la aplicación es la agilización en el aprendizaje y modificaciones de una serie de bloques de programación generados para llevar a cabo el control automático de ya puedan ser, plantas termosolares, plantas de bioetanol etc.

Es una herramienta de ayuda para el programador, utilizable en tiempo real, y que va a permitir que todo aquel que lleve a cabo

la programación de un determinado bloque, pueda explicar con todo lujo de detalles, dicho bloque cuando este sea creado, o todas las modificaciones que se lleven a cabo en estos a lo largo de su programación.

Además de esto, se busca la agilización en el proceso de validación de los bloques que se van a ir generando por parte de la persona responsable de los programadores, en el caso de estar hablando de la programación en oficina, o del jefe de puesta en marcha, en el caso de la carga en el sistema de control en campo de dichas plantillas.

Además de ser una herramienta de ayuda y control de las modificaciones para los programadores o aquellos que puedan consultar las diferentes plantillas, es una herramienta de control de cambios en los distintos proyectos en los que van a estar incluidas dichas plantillas, siendo este caso el perfil de jefe de puesta en marcha que será la persona que pueda incluir y acceder para modificar los sucesivos cambios en los distintos proyectos.

En el proceso de creación y modificación de una plantilla se distinguen dos partes diferenciadas hasta la utilización de ésta en planta. Una primera de programación en oficina, en la que el programador podrá incluir en la aplicación los sucesivos cambios, de forma que cuando otro programador necesite información al respecto de dicha plantilla, pueda ver todas las modificaciones y cambios, así como una descripción detallada de ésta.

En esta fase en oficina, se llevará a cabo el proceso de *1ª validación* de la plantilla, esto es, el responsable de cada programador, validará el correcto funcionamiento de ésta, para ello, la aplicación consta de un tercer tipo de gestión llamada *Gestión de reuniones*.

Este tercer bloque permite la creación por parte del programador con su responsable, de una reunión para validar o no las plantillas correspondientes. Todos los cambios que se lleven a cambio quedarán registrados en la base de datos, así como la reunión organizada por parte del programador.

Al igual que las reuniones de *1ª validación*, dentro de la Gestión de Reuniones existe la posibilidad de gestionar las reuniones de *2ª Validación*, que en este caso tendrán lugar en planta, entre el jefe de puesta en marcha y el programador que realizará la carga de los archivos correspondientes.

En cuanto a la estructura, se puede decir que la aplicación se encuentra dividida en cuatro bloques:

- Gestión de usuarios
- Gestión de Plantillas
- Gestión de Proyectos
- Gestión de Reuniones

En el primero bloque de gestión se encontrarán todos los archivos que corresponden a la generación de nuevas altas de usuarios, incluyendo el perfil de estos, bajas de usuarios, y modificación del perfil de estos. Todas estas opciones serán llevadas a cabo por el administrador de la aplicación, que tendrá acceso a todas las opciones del programa, y podrá modificar los distintos campos de las tablas que forman la base de datos, básica para el funcionamiento óptimo del sistema.

El segundo bloque de Gestión incluye todos los archivos correspondientes a la gestión de las distintas plantillas. Podrá insertarse una nueva en la base de datos, modificar, ver el listado completo o borrar, todo ello teniendo en cuenta que el perfil con el que se accede a la aplicación es el correcto para llevar a cabo cada una de estas acciones.

El bloque de Gestión de Proyectos incluye todos los archivos correspondientes a la gestión de los proyectos que se vayan incorporando. Es un bloque con un funcionamiento muy similar al de plantillas, variando la información a la que se accede, en este caso los proyectos, y el usuario que accede, modifica, consulta o borra dicha información.

Por último, el bloque de Gestión de Reuniones es el que permite la creación de las reuniones en la que se llevará a cabo la validación o no de las distintas plantillas.

Dentro de la aplicación, dependiendo del perfil con el que se acceda a ésta, esto es, *administrador*, *programador*, *responsable* o *jefe de puesta en marcha* se podrá utilizar las acciones correspondientes a dicho perfil.

En la siguiente pantalla se puede observar un ejemplo de la aplicación, en este caso, la organización de una reunión por parte del programador.

The screenshot displays the 'Gestor PPR' application interface. At the top, it says 'Gestor PPR' and 'Gestion de Programador'. Below this is a navigation menu with links: Home, Ayuda, Correo, Links, Forum, Actualizar, Descargas, Historico de Accesos, and logout. On the left side, there is a sidebar with a blue background containing a search box 'Resolucion de dudas:', a 'Contacto' link, and a 'Total Historico de accesos:' section with a list of months from Enero 2011 to Diciembre 2011. The main content area is white and contains several form fields: 'Fecha Creacion:' with a date picker set to '2011 - Mar', 'Nombre programador:' with a dropdown menu, a calendar view for March 2011, 'Nombre Jefe Puesta en Marcha:' with a dropdown menu set to 'Ignacio Suarez', and 'Nombre Responsable:' with a dropdown menu set to 'fernando pardo'. Below these are three input fields for 'Plantillas a Validar:' (value 20), '1ª Validacion:' (value 0), and '2ª Validacion:' (value 1). There are also fields for 'Hora reunion:' (value 16:00:00) and 'Sala Reunion:' (value Sala 2). A large text area labeled 'Descripcion' contains the following text: 'Se validarán las plantillas correspondientes teniendo en cuenta las últimas modificaciones llevadas a cabo por los directores del proyecto. Cualquier cambio deberá ser registrado debidamente en la aplicación. El documento debe ser cumplimentado por los presentes en la reunión y enviado a su respectivo responsable para dejar constancia de la reunión.' At the bottom of the form is a 'Confirmar reunion' button. The footer of the application reads 'Gestor de Plantillas, Proyectos y Reuniones by Blas Rodríguez Lerena'.

Figura 3.5.1.1 1

Ejemplo de la aplicación PPR

En los puntos siguientes se verá con detalle cada uno de los distintos bloques así como las opciones que ofrece cada uno de ellos.

3.6 Diseño y descripción de la base de datos

3.6.1 Análisis de la Base de Datos

En el siguiente apartado se mostrará el modelo E-R (Entidad – Relación) que representa a la Base de Datos del sistema y sobre la que tendrá que trabajar éste. Para realizar el modelo E-R, en primer lugar conviene observar y analizar la Base de Datos necesaria en el proyecto, para conocer qué datos almacena, cuáles no y cómo se relacionan entre sí.

En esta fase de análisis se mostrarán tanto el modelo E-R como el grafo relacional que definen a la Base de Datos. Una vez hecho esto, se mostrarán los diagramas de flujos de datos del sistema, para conocer de manera más concreta como se relacionan los datos dentro del mismo.

Hay que tener en cuenta que la aplicación no requiere de todas las entidades representadas en el siguiente modelo, por tanto a continuación se especifican cuáles de ellas son relevantes en este proyecto.

De los requisitos observados en el apartado 3.5.1 se observa que la aplicación solo requiere datos relacionados con plantillas, proyectos, reuniones y usuarios.

Pero estos datos en algunos casos están dispersos en diferentes entidades o no existen entidades propias para albergarlos.

Las entidades existentes con las que se relaciona este proyecto son:

POR PARTE DE LA PLANTILLA:

- Plantilla

POR PARTE DEL PROYECTO:

- Proyecto

POR PARTE DE LA REUNIÓN:

- Reunión

POR PARTE DEL USUARIO:

- Usuario

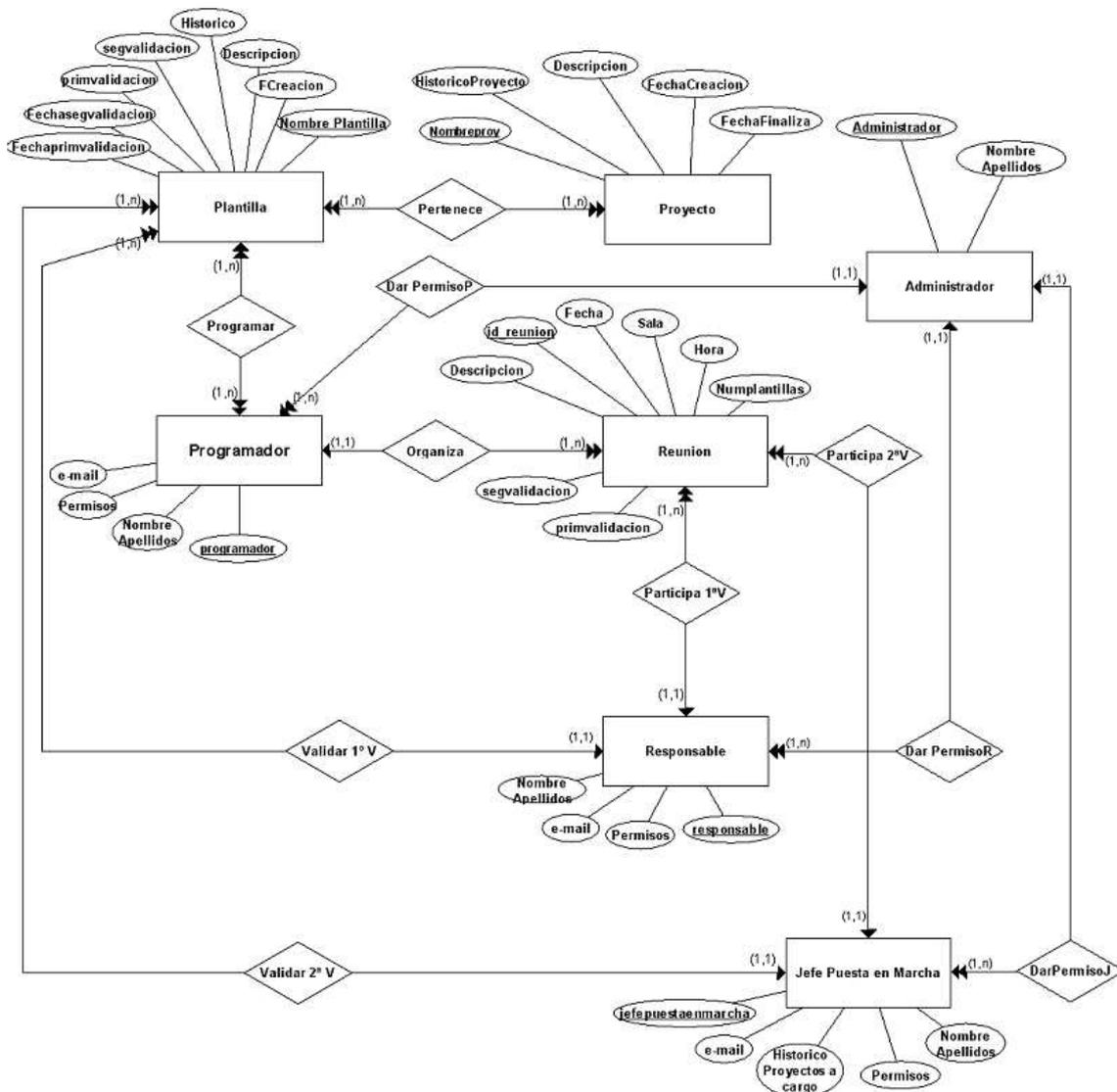


Figura 3.6.1.1 1

Modelo Entidad – Relación del Gestor PPR

De este modelo E-R se obtiene el siguiente grafo relacional donde se hace la transformación a tablas, se indican sus claves y relaciones, así como sus propiedades respecto a la integridad referencial.

3.6.2 Grafo Relacional

Del modelo relacional total se han separado las tablas junto las dependencias que afectan al proyecto, dejando de lado demás información no relevante para el diseño de esta aplicación.

En la figura 3.6.2.1 de la página siguiente se puede observar el grafo relacional obtenido a partir del modelo Entidad – Relacional de la figura 3.6.1.1.

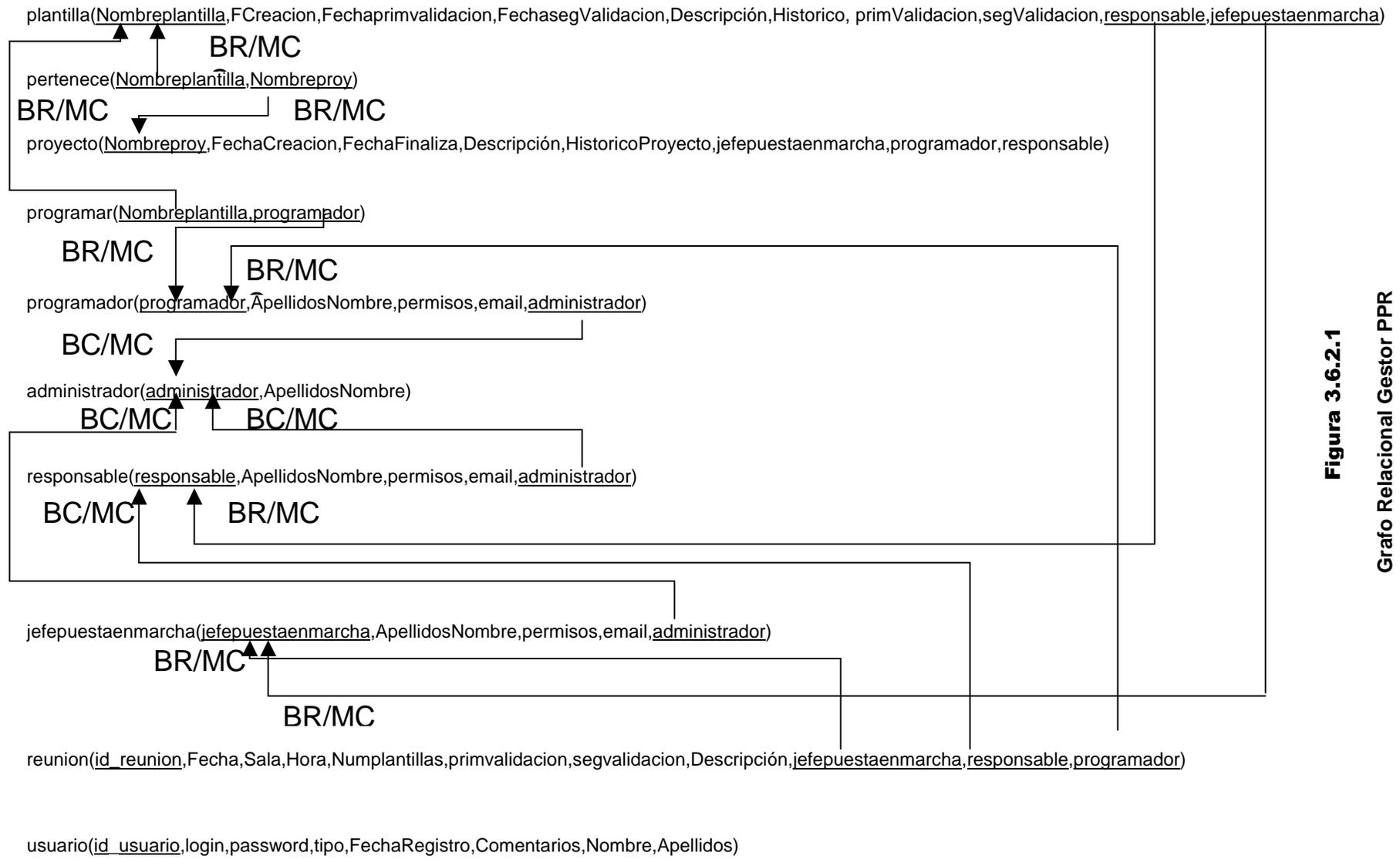


Figura 3.6.2.1

Grafo Relacional Gestor PPR

3.7 Especificación de Casos de Uso

Gracias a la especificación de procesos previamente detallados y a los requisitos se realizan los diferentes casos de uso. Agrupando en diferentes usos las actividades detalladas según el subsistema al que pertenezcan.

3.7.1 Gestión de Plantillas

Este caso de uso refleja las diferentes actividades o acciones que puede realizar el usuario para el subsistema de gestión de plantillas.

La aplicación tendrá la funcionalidad de incluir plantillas nuevas totalmente desde cero.

El usuario también tendrá un método que le permita buscar plantillas de manera concreta y rápida, con diferentes criterios de filtrado para su búsqueda, esto es, el programador que creó dicha plantilla, el responsable que la validó, su fecha de creación etc...

Además el usuario podrá visualizar todos los datos de una plantilla previamente registrado en la Base de Datos y en caso de estimarlo oportuno podrá modificar todos los datos que crea conveniente.

La aplicación también podrá realizar el borrado completo e indefinido de los datos de una plantilla almacenada en la Base de Datos.

La aplicación permite la visualización completa de todas las plantillas existentes, así como la posibilidad de exportar dicha lista en un archivo en formato Excel, el cual contendrá todos los datos específicos de cada una de las plantillas. En dichas plantillas se podrá ver quién fue el programador de cada una de ellas, cuándo se creó, etc...

En la siguiente figura se representan todos los casos de uso relacionados con la gestión de las plantillas. Como actor participa el usuario o el administrador de la aplicación, como ocurrirá con todos los casos de uso del proyecto.

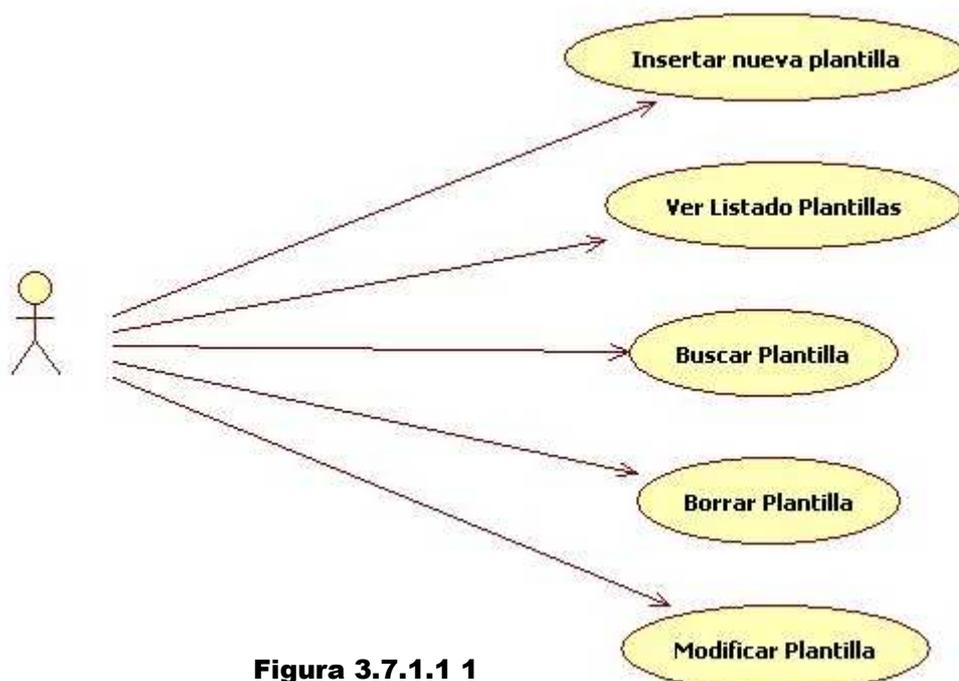


Figura 3.7.1.1 1
Gestión Plantillas

3.7.2 Gestión de Proyectos

Este caso de uso indica las actividades que el usuario puede realizar para el subsistema de gestión de proyectos.

El sistema debe facilitar la funcionalidad que permita buscar proyectos ya registrados dentro de la Base de Datos o crear uno nuevo desde 0. Se podrán determinar los criterios de filtrado que se consideren más convenientes para obtener el proyecto buscado, esto es, nombre, fecha de registro, fecha de finalización, programador, responsable etc...

Para los proyectos que se encuentren ya registrados se podrán consultar o editar sus datos, pudiendo modificar el campo que se desee.

La aplicación también permitirá el borrado del proyecto seleccionado de la Base de Datos, pudiendo seleccionar el proyecto dependiendo del criterio de búsqueda que se aplique.

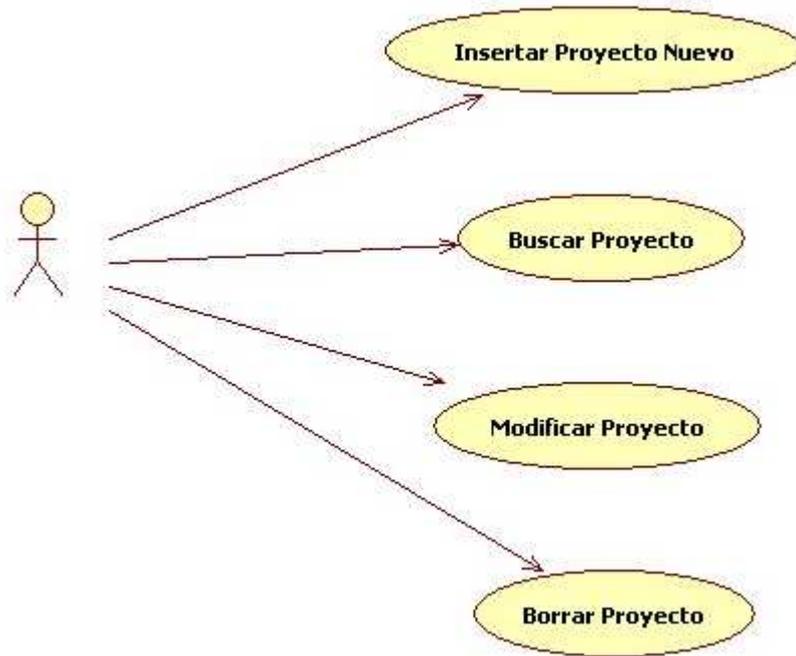


Figura 3.7.1.1 2
Caso de Uso Gestión Proyectos

3.7.3 Gestión de Reuniones

Este caso de uso indica las actividades que el usuario puede realizar para el subsistema de gestión de reuniones.

El sistema debe facilitar la funcionalidad que permita mostrar las reuniones pendientes por parte del usuario, mostrando un listado de dichas reuniones.

La aplicación permitirá la organización por parte de un usuario de una reunión, así como el borrado de la reunión seleccionada de la Base de Datos de manera indefinida.

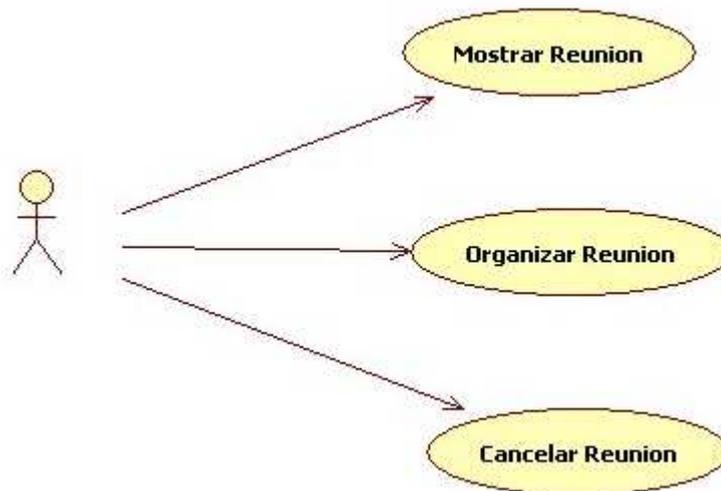


Figura 3.7.3.1 1
Caso de Uso Gestion Reuniones

3.7.4 Gestión Global

Este caso de uso muestra la actividad realizada por el usuario al acceder a la aplicación. El sistema deberá controlar el acceso de inicio, teniendo en cuenta el perfil del usuario que accede a la aplicación.

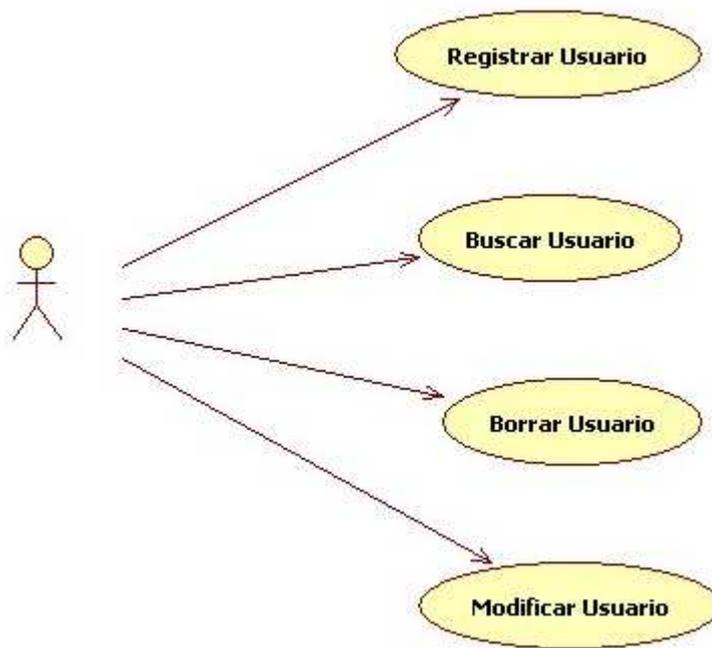


Figura 3.7.4.1 1
Caso de Uso Gestion Global

3.8 Análisis de los Casos de Uso

A través de analizar los casos de uso anteriores se pueden identificar las clases que participarán en el sistema. Una vez identificadas las clases participantes se realizará en diagrama de interacción entre ellas (diagrama de clases) que facilitará en gran medida la comprensión del funcionamiento del sistema.

3.8.1 Identificación de Clases Asociadas

El sistema se diferencia claramente en varios apartados según su uso. Este sistema, como se puede comprobar en los

casos de uso, debe presentar varias clases primordiales para el control de las funcionalidades requeridas, estas son:

- **Plantilla:** Clase encargada de la gestión de las plantillas a tratar.
- **Proyecto:** Clase encargada del control y gestión de los proyectos del sistema.
- **Reunión:** Clase encargada del control y gestión de las reuniones.
- **Usuario:** Clase encargada de validar la entrada del usuario en el sistema.
- **Sistema de Gestión:** Clase central que gestiona el resto de clases y por consiguiente, el sistema.

3.9 Modelo Estático de Datos: Diagrama de Clases

Por medio del diagrama de clases, que es un tipo de diagrama estático, se va a describir la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. Este diagrama se ha realizado durante el proceso de análisis y diseño de la aplicación, de esta forma se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

Para realizar las funcionalidades del sistema se requieren un conjunto de clases, mediante las cuales el sistema tratará los datos almacenados en la Base de Datos.

Como podrá observarse, el diagrama de clases no es nada complejo, pero a pesar de ello sí que contiene un gran número de datos y operaciones posibles que manejar.

A continuación se muestran las clases del sistema y el grado de instancias relacionadas que presentan entre sí, esto es, la cardinalidad.

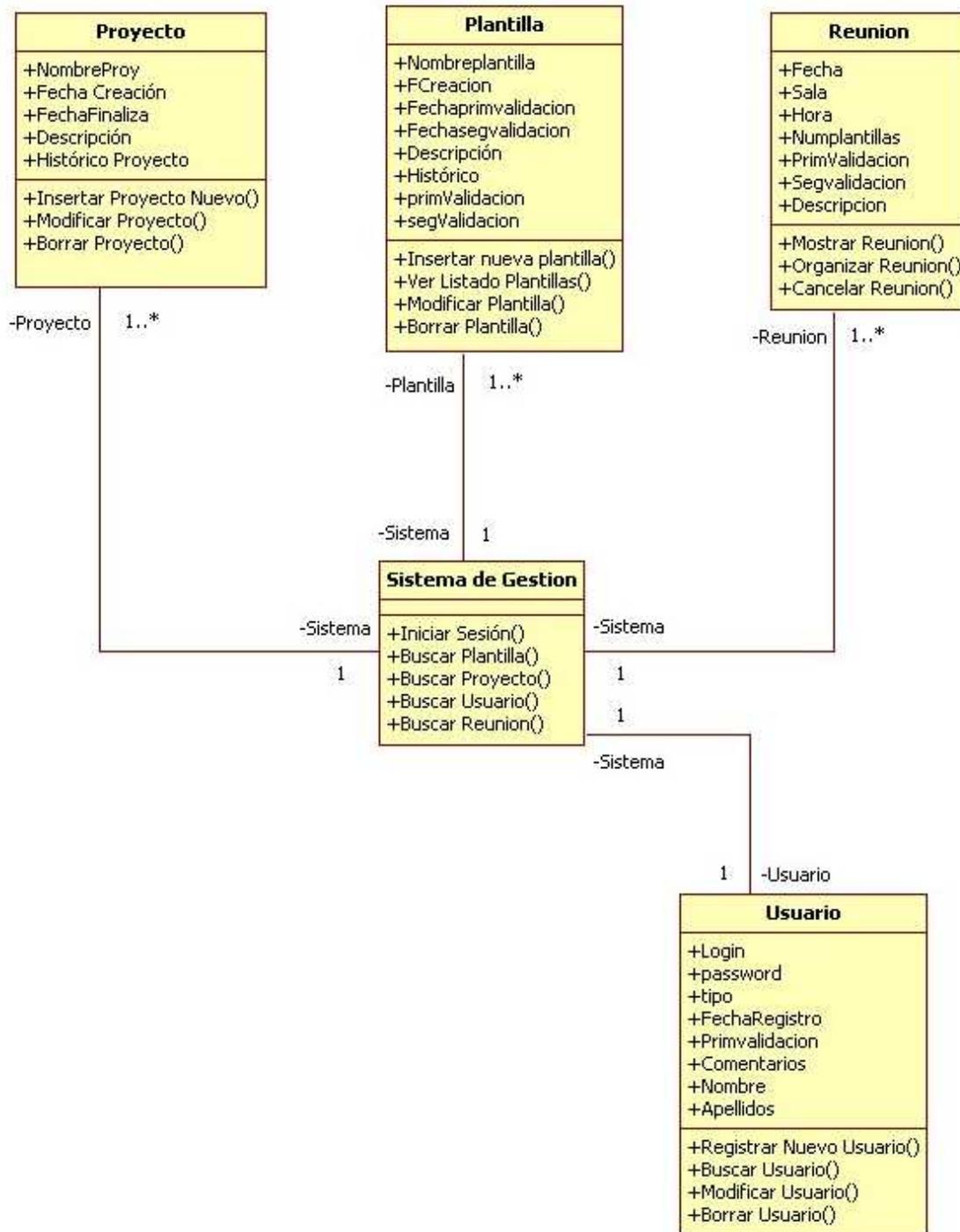


Figura 3.9.1 1
Diagrama de Clases

3.10 Análisis de Clases

Se definen cada una de las clases junto con sus atributos, funcionalidad y responsabilidades pertenecientes al sistema.

Plantilla	
Responsabilidades	Clase sobre la que se gestiona a las plantillas almacenadas en el sistema.
Atributos	<ul style="list-style-type: none">• Nombreplantilla: Nombre de la plantilla. Atributos que identifican de manera unívoca a la plantilla.• FCreación: Fecha de creación de la plantilla• Fechaprimvalidación: Indica la fecha en que se realiza la 1ª validación de la plantilla por parte del Responsable.• Fechasegvalidación: Indica la fecha en que se realiza la 2ª validación de la plantilla por parte del Jefe de Puesta en Marcha.• Descripción: Descripción detallada de la plantilla, indicando su función dentro del proyecto.

	<ul style="list-style-type: none">• Histórico: Se indican los proyectos en los que ha sido utilizada la plantilla, así como los cambios o modificaciones que se hayan realizado sobre ésta.• primValidación: Indica si está o no validada por primera vez la plantilla.• segValidación: Indica si está o no validada por segunda vez la plantilla.
<p>Métodos</p>	<ul style="list-style-type: none">• Insertar nueva plantilla: Registra la plantilla en el sistema.• Ver Listado Plantillas: Muestra un listado de todas las plantillas.• Borrar Plantilla: Elimina del sistema los datos de la plantilla indicada.• Modificar Plantilla: Modifica la información de la plantilla seleccionada.

Análisis de Clase Plantilla

Proyecto	
Responsabilidades	Clase sobre la que se gestiona a los proyectos almacenados en el sistema.
Atributos	<ul style="list-style-type: none">• NombreProy: Nombre del proyecto. Describe de manera única a un proyecto dentro del sistema.• Fecha Creación: Fecha de creación del proyecto.• FechaFinaliza: Fecha de finalización del proyecto, ésta será dada por el jefe de puesta en marcha.• Descripción: Descripción detalle del proyecto.• Histórico Proyecto: Almacena todos los cambios reseñables que tienen lugar en el proyecto.
Métodos	<ul style="list-style-type: none">• Insertar Proyecto Nuevo: Registra un nuevo proyecto en el sistema.• Modificar Proyecto: Modifica cualquier tipo de información del proyecto.• Borrar Proyecto: Elimina

	del sistema los datos referentes al proyecto indicado.
--	--

Análisis de Clase Proyecto

Reunión	
Responsabilidades	Clase sobre la que se gestiona a las reuniones almacenadas en el sistema.
Atributos	<ul style="list-style-type: none">• Fecha: Fecha fijada para la reunión.• Sala: Indica la sala en la que tendrá lugar la Reunión de Validación.• Hora: Indica la hora en que tendrá lugar la Reunión de Validación.• Numplantillas: Número de plantillas que se pretenden validar en la Reunión.• Primvalidación: Indica si se va a realizar la primera Validación.• Segvalidación: Indica si se va a realizar la segunda validación de la plantilla en la reunión.• Descripción: Descripción de la reunión que tendrá

	<p>lugar y los puntos que se seguirán en ésta.</p>
Métodos	<ul style="list-style-type: none"> • Mostrar Reunión: Muestra en el sistema las reuniones pendientes según los criterios de selección seguidos. • Organizar Reunión: Almacena en el sistema la reunión creada. • Cancelar Reunión: Elimina del sistema los datos referentes a la reunión creada.

Análisis de Clase Reunión

Usuario	
Responsabilidades	<p>Clase que contendrá los datos referentes a los usuarios que empleará el sistema.</p>
Atributos	<ul style="list-style-type: none"> • Login: Nombre utilizado por el usuario para acceder a la aplicación. • password: Palabra clave a escribir en el comienzo de la aplicación para validar que sólo el usuario podrá

	<p>acceder al sistema.</p> <ul style="list-style-type: none">• tipo: Indica el tipo de usuario que va a acceder al sistema.• FechaRegistro: Indica la fecha en que se dio de alta al usuario en la aplicación por parte del administrador.• Comentarios: Se indica cualquier comentario referente al usuario, los cambios de perfil que ha podido sufrir etc.• Nombre: Nombre del usuario.• Apellidos: Apellidos del usuario.
<p>Métodos</p>	<ul style="list-style-type: none">• Registrar Nuevo Usuario: Inclusión de un nuevo usuario por parte del administrador en la base de datos del sistema.• Modificar Usuario: Modifica la información referente a un usuario incluido en la base de datos del sistema.• Borrar Usuario: Elimina del

	sistema los datos referentes al usuario perteneciente a éste.
--	---

Análisis de Clase Usuario

Sistema de Gestión	
Responsabilidades	Clase central del sistema, llevará un control de los diferentes procesos y envíos de datos entre las clases del propio sistema. También maneja las interfaces mediante las que el usuario comunicará con el sistema.
Atributos	
Métodos	<ul style="list-style-type: none">• Iniciar Sesión: Entra en el sistema según las interfaces adecuadas.• Buscar Plantilla: Busca una plantilla dentro del sistema.• Buscar Proyecto: Busca un proyecto en el sistema según los criterios de búsqueda especificados.• Buscar Usuario: Busca un determinado usuario en la base de datos del sistema.

Análisis de Clase Sistema de Gestión

3.11 Descripción de Interacción. Diagramas de Secuencia

Los diagramas escogidos para representar la interacción son los de secuencia. Se representa la interacción entre objetos del sistema según las principales tareas que se van a poder realizar.

3.11.1 Diagrama de Secuencia Modificar Plantilla

La operación a diagramar será la producida cuando un usuario pretende modificar algún aspecto de una plantilla ya registrada en el sistema.

El usuario envía a la clase sistema de gestión el nombre de usuario y password para identificarse, después de validarse la contraseña y ser correcta, se busca la plantilla, se carga esa plantilla con la información ya existente, y finalmente se envían los datos modificados antes de registrar los cambios.

Los diagramas de secuencia modificar usuario y modificar proyecto son idénticos al aquí mostrado, difieren únicamente en la información a modificar de la base de datos, por tanto, no son mostrados.

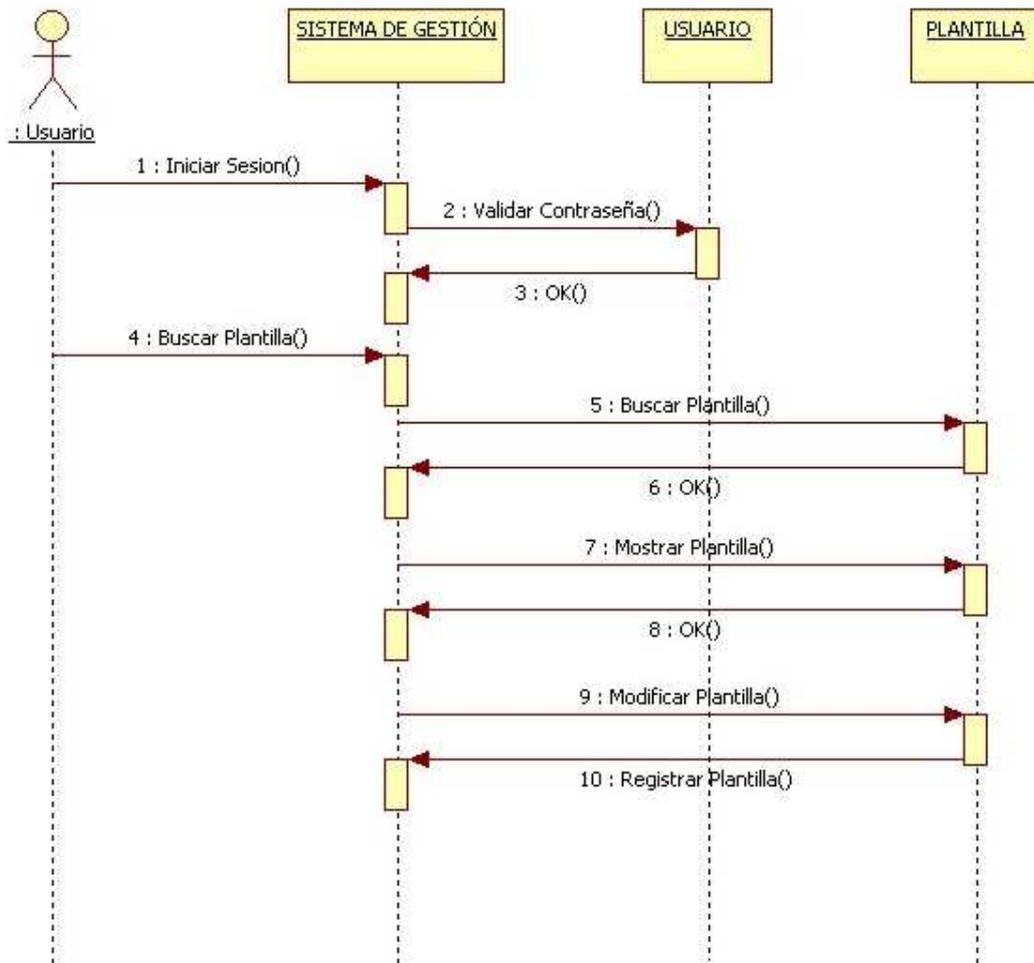


Figura 3.11.1.1 1

Diagrama de Secuencia Modificar Plantilla

3.11.2 Diagrama de Secuencia Buscar Plantilla

La operación de la que se representa el diagrama de secuencia, es aquella en la que el usuario pretende buscar una determinada plantilla en la aplicación, según los criterios de búsqueda que especifique.

El usuario envía a la clase sistema de gestión la plantilla a buscar, con la información necesaria de ésta, la clase sistema de gestión envía a su vez una petición a la clase plantilla para mostrar sus datos. Una vez hecho esto, se mostrará la información de la plantilla mediante la interfaz adecuada, mediante la operación mostrar plantilla.

Los diagramas de secuencia de buscar proyecto y mostrar reuniones serán idénticos al aquí mostrado, por tanto no se muestran.

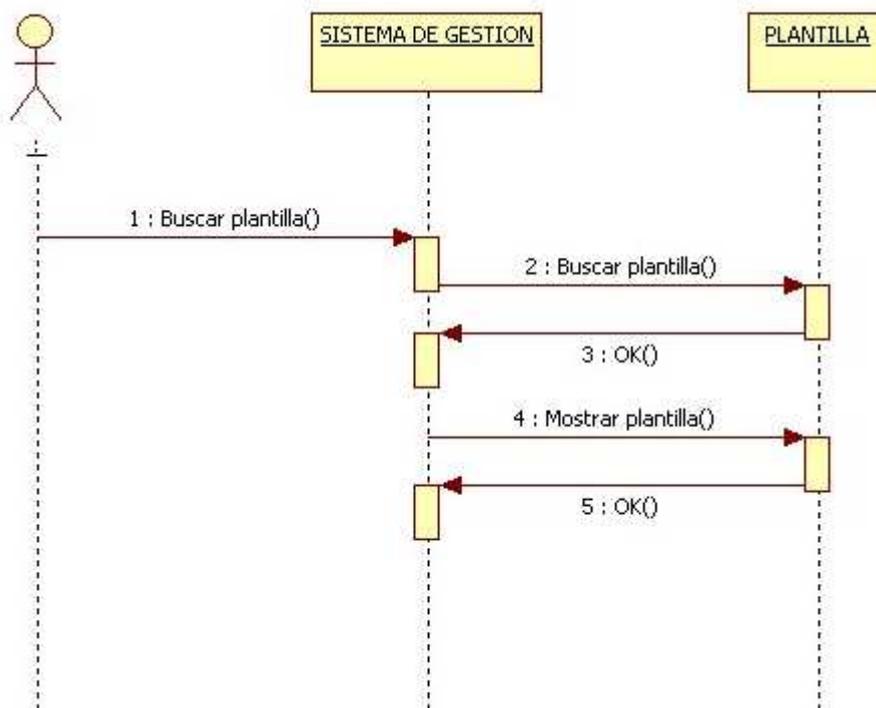


Figura 3.11.2.1 1

Diagrama de Secuencia Buscar Plantilla

3.11.3 Diagrama de Secuencia Borrar Plantilla

Se muestra el diagrama de secuencia para el caso en que un usuario desee borrar una plantilla del sistema.

Se parte con un usuario correctamente conectado al sistema. Posteriormente el usuario envía el mensaje de buscar una plantilla concreta al sistema de gestión, el cual se encarga de pasar el mensaje a la clase plantilla para que lo muestre.

Una vez hecho esto el usuario indicará a la aplicación que desea borrar la plantilla que se está visualizando. A continuación la clase plantilla borra los datos de la plantilla a los que tiene acceso.

Finalizadas correctamente las actividades, las clases se envían entre sí mensajes de confirmación.

Los diagramas de secuencia borrar proyecto y cancelar reunión son idénticos al a continuación mostrado, de tal forma no son representados en la memoria del proyecto.

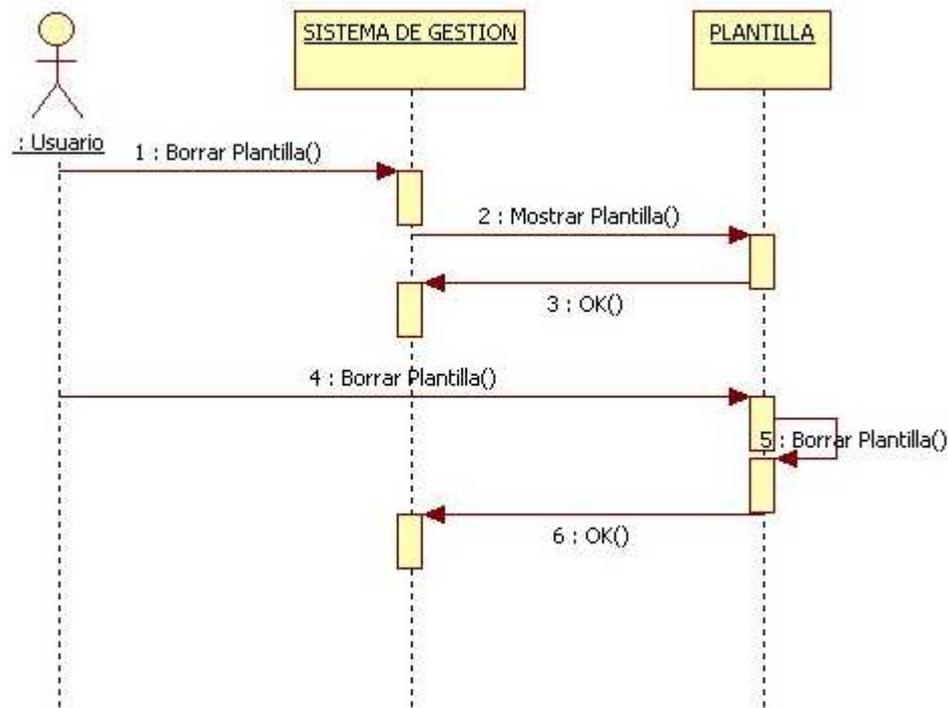


Figura 3.11.3.1 1

Diagrama de Secuencia Borrar Plantilla

3.11.4 Diagrama de Secuencia Ver Listado Plantillas

A continuación se muestra el diagrama de secuencia para el caso en que un usuario desee ver por pantalla todo el listado plantillas existente.

Se ha tenido en cuenta que el usuario todavía no ha accedido a la aplicación, incluyéndose por tanto como accede el usuario a ésta. El usuario envía a la clase sistema de gestión el nombre de usuario y contraseña para identificarse.

Una vez que se ha validado correctamente, la clase sistema de gestión envía a la clase plantilla un mensaje para que muestre el listado de plantillas. Al recibir una respuesta afirmativa, el sistema de gestión muestra en pantalla el listado de plantillas existentes.

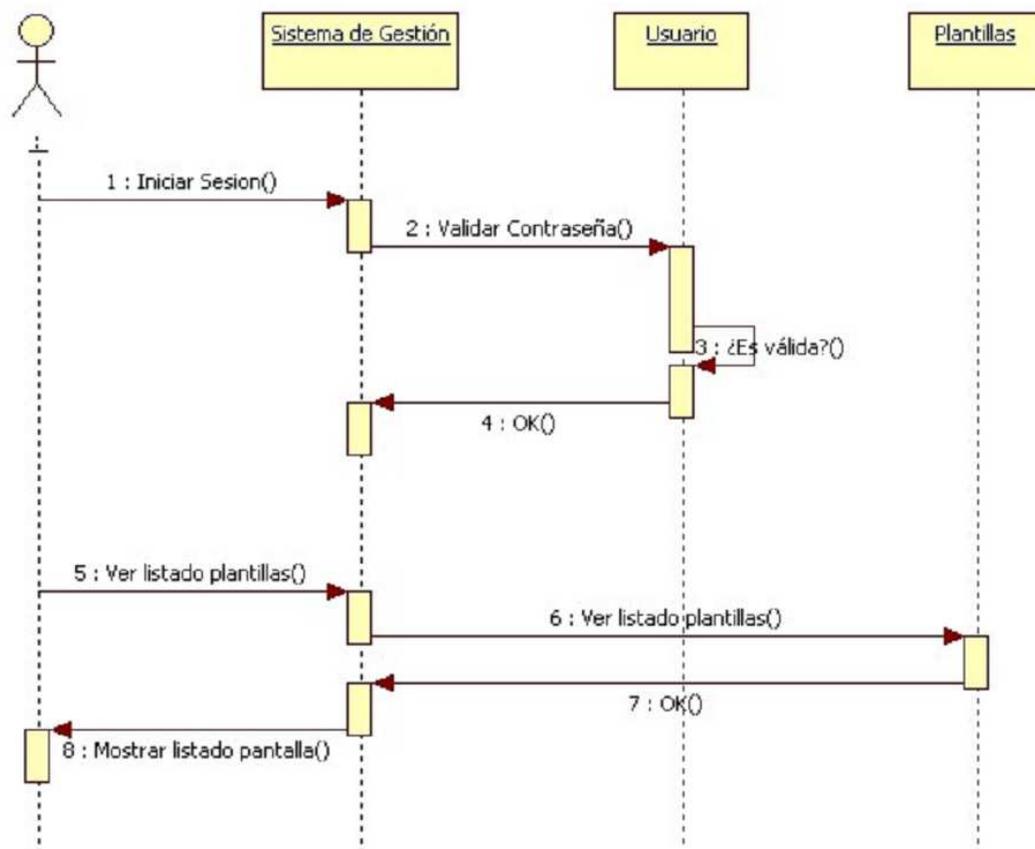


Figura 3.11.4.1 1

Diagrama de Secuencia Ver Listado Plantillas

3.11.5 Diagrama de Secuencia Organizar Reunion

Este diagrama describe la secuencia que se da cuando un usuario programador organiza una reunión de primera validación.

El usuario envía a la clase sistema de gestión el nombre de usuario y contraseña para identificarse, después de validarse la contraseña en la clase usuario y ser correcta, el usuario envía a la clase sistema de gestión la orden organizar reunion, el sistema de gestión a su vez se la manda a la clase reunion.

El usuario manda la orden registrar datos reunion con todos los datos correspondientes a la nueva reunion, este a su vez los manda a la clase reunión, la cual mandará la señal de que la reunión es correcta a la clase sistema de gestión, finalmente se registrarán los datos de la reunión y se dará el visto bueno a la operación.

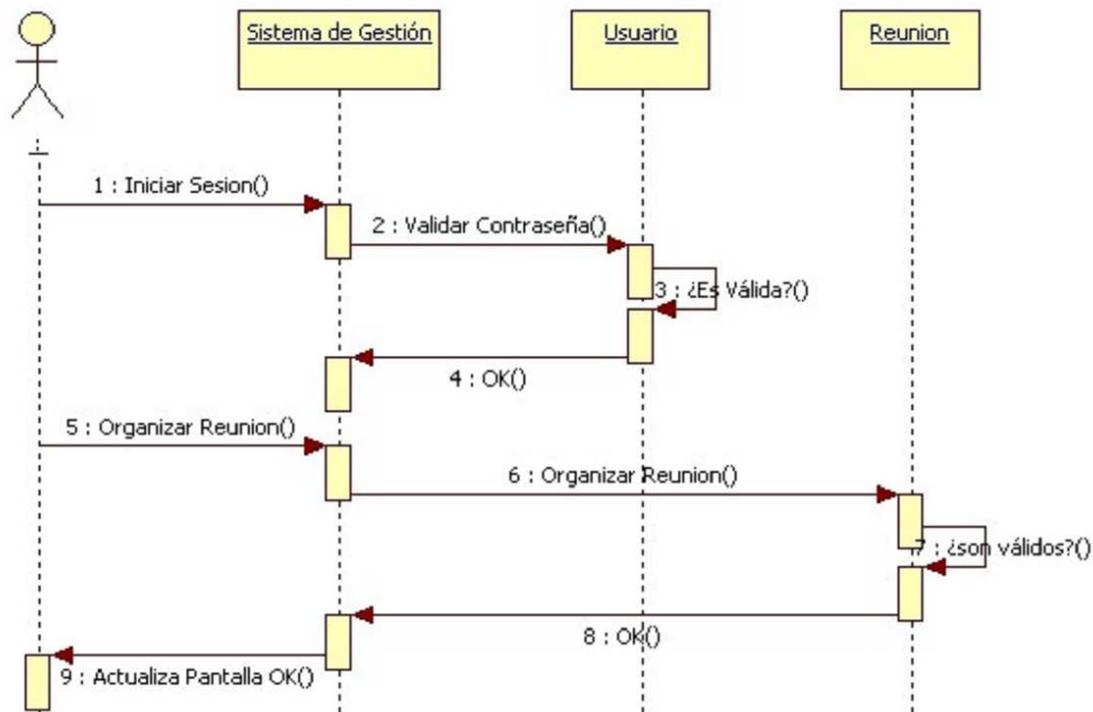


Figura 3.11.5.1 1

Diagrama de Secuencia Organizar Reunión

3.11.6 Diagrama de Secuencia Insertar Proyecto Nuevo

La operación a diagramar va a ser la producida cuando un usuario registra un proyecto nuevo en la base de datos de la aplicación. En este caso se parte de que el usuario está ya identificado en el sistema, por tanto no va a aparecer en el diagrama la identificación de dicho usuario.

La clase sistema de gestión envía un mensaje a las clase proyecto para insertar el nuevo proyecto, una vez que la respuesta sea afirmativa, el sistema de gestión muestra en pantalla un mensaje de que la inserción se realizó correctamente.

Los diagramas de secuencia registrar nuevo usuario e Insertar plantilla nueva son idénticos al aquí representado, difiriendo sólo en la información nueva a añadir en la base de datos, de forma que no aparecen representados.

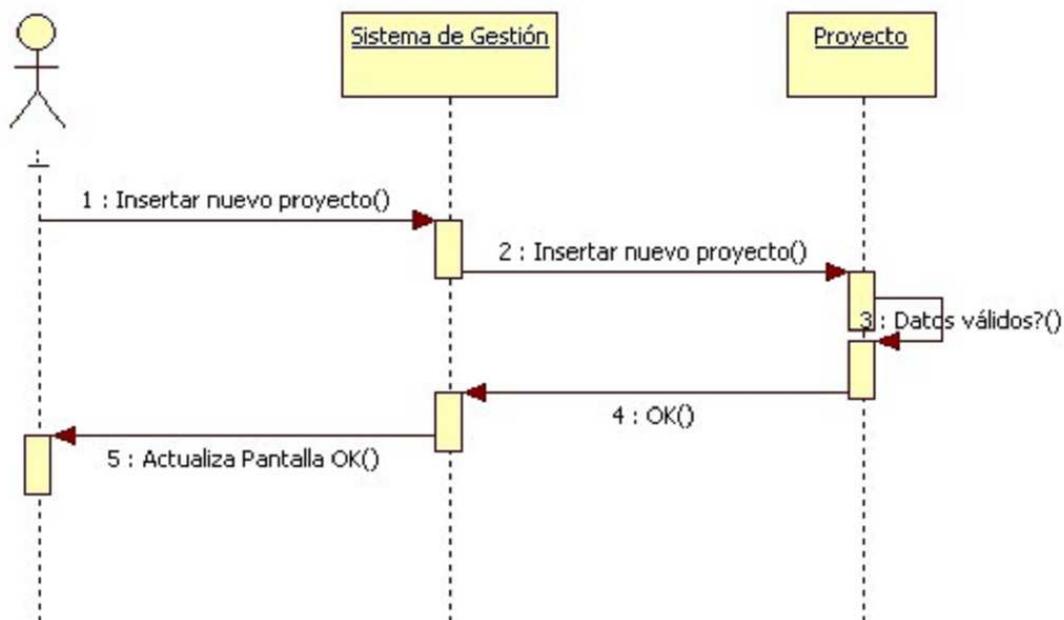


Figura 3.11.6.1 1

Diagrama de Secuencia Insertar Proyecto Nuevo