



**Proyecto de Fin de Carrera**

**Universidad Politécnica de Valencia**

**Escuela Técnica Superior de Informática**

***Desarrollo de un Sistema de  
Gestión de Proyectos  
mediante el framework GWT***

Escuela Técnica Superior de Ingeniería Informática 

***Realizado por:***  
**Reyes Martínez Durbán**

***Dirigido por:***  
**D. Moisés Pastor i Gadea**

*Valencia, Junio 2012*

## Índice

1. DESCRIPCIÓN.....	2
1.1. MOTIVACIÓN.....	2
1.2. OBJETIVOS.....	3
2. ANÁLISIS DE REQUISITOS.....	3
2.1. REQUISITOS DE INFORMACIÓN.....	5
2.2. REQUISITOS FUNCIONALES.....	8
2.1.1. DIAGRAMAS DE CASOS DE USO.....	8
2.2.2. DEFINICIÓN DE ACTORES.....	9
2.2.3. DESCRIPCIÓN TEXTUAL DE CASOS DE USO.....	9
2.3. REQUISITOS NO FUNCIONALES.....	33
3. APLICACIÓN.....	33
3.1. PROCESO DE INSTALACIÓN.....	33
3.2. BASE DE DATOS.....	35
3.3. DESCRIPCIÓN DE LA APLICACIÓN.....	36
3.4. PRUEBAS.....	41
4. CONCEPTOS TEÓRICOS .....	42
4.1. AJAX.....	42
4.1.1. DEFINICIÓN AJAX.....	42
4.2. GWT.....	43
4.2.1. DEFINICIÓN GWT.....	43
4.2.1.1. VENTAJAS.....	44
4.2.1.2. INCONVENIENTES.....	44
4.2.2. ARQUITECTURA DE GWT.....	44
4.2.3. INTERFAZ DE USUARIO.....	45
4.2.4. EL PATRÓN MODELO-VISTA-PRESENTADOR.....	50
4.2.4.1. DEFINICIÓN.....	50
4.2.4.2. IMPLEMENTACIÓN DEL PATRÓN MVP: ACTIVITIES Y PLACES.....	50
4.2.5. COMUNICACIÓN CON EL SERVIDOR .....	52
4.2.5.1. LLAMADAS RPC .....	53
4.2.5.2. MODOS DE EJECUCIÓN.....	56
5. CONCLUSIONES.....	56
6. BIBLIOGRAFÍA.....	56

## 1. DESCRIPCIÓN

El proyecto a desarrollar consiste en una aplicación Web cuya finalidad será la de administrar y controlar los distintos proyectos desarrollados por una organización.

Además, pretende cubrir la necesidad de establecer un seguimiento sobre las tareas desarrolladas por los distintos técnicos, con lo cual se cumple una doble finalidad:

La primera es el control de las propias tareas en las que se descomponen los distintos módulos de cada proyecto de la organización y la segunda consiste en un control exhaustivo de las tareas desarrolladas por los distintos técnicos de la organización y el tiempo empleado en cada una de ellas.

Por tanto, la aplicación puede proporcionar un mecanismo muy útil a la hora de realizar estimaciones para proyectos futuros, en cuanto a coste, complejidad, duración, etc., así como la posibilidad de medir el rendimiento y la productividad de los técnicos, que puede permitir a la organización optimizar futuros proyectos.

El proyecto se divide cinco partes funcionales, las cuales serían:

**Gestionar los Proyectos:**

El sistema deberá gestionar toda la información referente a los proyectos software desarrollados por una organización. Por lo tanto se contemplarán tareas de gestión como: alta de proyectos en el sistema, baja de uno o varios de ellos, modificaciones sobre los datos de cada uno, consultas, etc.

**Gestionar los Módulos:**

El sistema deberá gestionar la información del conjunto de módulos que constituyen cada uno de los proyectos del sistema. Por tanto debe permitir: añadir módulos a proyectos, borrar módulos de un proyecto determinado, modificaciones de los datos de un módulo, consultas, etc.

#### Gestionar las Tareas:

El sistema deberá gestionar toda la información relativa a las tareas que se realizan para construir cada uno de los módulos de los proyectos software. Por tanto debe incluir actividades como: dar de alta tareas y asociarlas a módulos, especificar el/los técnico(s) que van a ejecutar cada tarea y cuándo la harán, modificar datos de tareas existentes, consultas, etc.

#### Gestionar los Técnicos:

El sistema deberá gestionar los técnicos susceptibles de ser asignados a las tareas de los módulos de los proyectos software, ya sea como responsable o como desarrollador. Por tanto deberá incluir tareas como: alta de técnicos, baja, modificaciones, consultas, asignación a tareas de módulos de proyectos, mantenimiento, etc.

#### Gestionar las Asignaciones de Técnicos a Tareas:

El sistema deberá poder consultar el conjunto de actividades, desarrolladas por cada técnico asociado a alguno de los proyectos del sistema, en un periodo de tiempo establecido por el usuario bajo unos criterios específicos: técnicos de un proyecto dado, técnicos por módulo de un proyecto dado, técnicos por tarea de un proyecto dado, todos los técnicos del sistema o un técnico en concreto.

## **1.1. MOTIVACIÓN**

La idea de realizar este proyecto final de carrera surgió como consecuencia del trabajo desarrollado en una empresa de desarrollo de software, en el cual estuve desarrollando aplicaciones utilizando la tecnología utilizada en el proyecto y la que considero puntera y demandada en el mercado actual por su fácil mantenimiento y su flexibilidad a la hora de realizar cambios.

Después de haber utilizado GWT en varios proyectos y tener los conocimientos suficientes para desarrollar una aplicación con esta tecnología, me decanté por ésta para aprovechar los conocimientos adquiridos durante este periodo.

## **1.2. OBJETIVOS**

1. Ser capaz de descargar, instalar y configurar el software necesario para la aplicación.
2. Aprender a desarrollar portales Web con GWT.
3. Aprender a instalar el servidor Apache Tomcat.
4. Aprender a instalar el servidor de BBDD MySQL.
5. Desarrollar un portal web sobre la gestión de proyectos.
6. Ampliar el conocimiento del entorno de desarrollo Eclipse.
7. Ampliar los conocimientos sobre patrones de diseño.
8. Mejorar la capacidad de comunicación escrita debido a la realización de la memoria del PFC.
9. Aumentar la capacidad de autoformación, tanto de búsqueda de información como de comprensión de los distintos manuales utilizados.

## **2. ANÁLISIS DE REQUISITOS**

El proyecto “Sistema de Gestión de Proyectos” está estructurado de la siguiente manera: los proyectos son los sistemas o aplicaciones a desarrollar en la organización, así como los trabajos globales.

Se componen de módulos y cada módulo se descompone a su vez en tareas más concretas. Una tarea la ejecutan uno o varios técnicos, si bien existe un técnico responsable en cada proyecto.

Tanto el proyecto como el módulo y cada tarea tienen fecha de comienzo y fecha de finalización.

Las tareas son realizadas por técnicos, los cuales tienen asignada una tarea durante un periodo de tiempo.

Una vez descrito brevemente como sería la estructura de la aplicación, los puntos fundamentales que se abordarían en ella serían los siguientes:

Referente a proyectos:

1. Alta de nuevos proyectos
2. Consulta de proyectos
3. Modificación de proyectos
4. Baja de proyectos

Referente a técnicos:

1. Alta de nuevos técnicos
2. Consulta de técnicos
3. Modificación de técnicos
4. Baja de técnicos

Referente a módulos:

1. Alta de nuevos módulos
2. Consulta de módulos
3. Modificación de módulos
4. Baja de módulos

Referente a tareas:

1. Alta de nuevas tareas
2. Consulta de tareas
3. Modificación de tareas
4. Baja de tareas

Referente a asignaciones:

1. Alta de nuevas asignaciones
2. Consulta de asignaciones
3. Modificación de asignaciones
4. Baja de asignaciones

Tareas Combinadas:

1. Añadir módulos a proyectos
2. Suprimir módulos de proyectos
3. Añadir tareas a módulos
4. Suprimir tareas de módulos
5. Asignación de técnicos responsables a proyectos
6. Suprimir asignaciones de técnicos responsables a proyectos
7. Consultas de asignaciones por proyecto, módulo, tarea o técnico, tanto activos como dados de baja.

## **2.1. REQUISITOS DE INFORMACIÓN**

En este apartado se identifican los requisitos de almacenamiento de información que debe cumplir el sistema. Estos requisitos son los que deben contestar a la pregunta: “¿Qué información, relevante para los objetivos de negocio, debe ser almacenada por el sistema?”. Esta información necesaria para los objetivos propuestos sería:

El sistema debe almacenar la información correspondiente a los proyectos software. En concreto:

- Identificador del proyecto (clave primaria en la BBDD)
- Nombre
- Descripción

- Alcance
- Objetivos
- Origen
- Coste total
- Técnico responsable (jefe de proyecto)
- Fecha de inicio del desarrollo
- Fecha de finalización del desarrollo

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- El identificador del proyecto debe ser único, por tanto, no podrán existir dos proyectos distintos con el mismo identificador.
- El identificador del proyecto es un campo autogenerado por la BBDD al crear el proyecto, por tanto, no puede ser modificado.
- Los campos Nombre, Coste y Técnico Responsable del proyecto no pueden tomar valores nulos.

El sistema deberá almacenar la información correspondiente a los módulos o subsistemas que componen cada proyecto software. En concreto:

- Identificador de módulo (clave primaria en la BBDD)
- Proyecto al que pertenece (clave ajena en la BBDD)
- Nombre
- Descripción
- Fecha de inicio del desarrollo
- Fecha de finalización del desarrollo
- Observaciones

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- El identificador del módulo debe ser único, por tanto, no podrán existir dos módulos distintos con el mismo identificador.
- El identificador del módulo es un campo autogenerado por la BBDD al crear el módulo, por tanto, no puede ser modificado.
- Los campos Nombre y Proyecto al que pertenece no pueden tomar valores nulos.

El sistema deberá almacenar la información correspondiente a las tareas a realizar en el desarrollo de cada uno de los módulos de los proyecto software. En concreto:

- Identificador de tarea (clave primaria en la BBDD)
- Módulo al que pertenece (clave ajena en la BBDD)
- Nombre
- Descripción
- Fecha de inicio del desarrollo
- Fecha de finalización del desarrollo
- Observaciones

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- El identificador de la tarea debe ser único, por tanto, no podrán existir dos tareas distintas con el mismo identificador.
- El identificador de la tarea es un campo autogenerado por la BBDD al crear la tarea, por tanto, no puede ser modificado.
- Los campos Nombre y Módulo al que pertenece no pueden tomar valores nulos.

El sistema deberá almacenar la información correspondiente a los técnicos implicados en cada proyecto software. En concreto:

- NIF (clave primaria en la BBDD)
- Nombre
- Apellidos
- Puesto en la organización (Clave ajena en la BBDD)
- Fecha de antigüedad
- Fecha de baja

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- El identificador (NIF) del técnico debe ser único, por tanto, no podrán existir dos técnicos distintos con el mismo NIF.
- Los campos Nombre, Apellidos, NIF y Puesto no pueden tomar valores nulos.

El sistema deberá almacenar la información correspondiente a los técnicos que llevan a cabo cada tarea en el sistema. En concreto:

- Tarea que ejecuta el técnico (Clave primaria en la BBDD)
- NIF del técnico que ejecuta la tarea (Clave primaria en la BBDD)
- Fecha de asignación de la tarea
- Fecha de salida de la tarea

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- En el sistema no se puede registrar al mismo técnico la misma tarea dos veces. Es decir, un técnico ejecuta cada tarea una única vez.
- Los campos Tarea y Técnico no pueden tomar valores nulos.

El sistema deberá almacenar la información correspondiente a los diferentes perfiles que un técnico puede tener dentro de la organización. En concreto:

- Identificador del perfil (Clave primaria en la BBDD)
- Descripción del perfil

La información almacenada por el sistema deberá satisfacer las siguientes restricciones:

- El identificador del perfil debe ser único, por tanto, no podrán existir dos perfiles distintos con el mismo identificador.
- El campo Descripción no puede tomar valores nulos.
- Se trata de datos estáticos, no modificables por el usuario. Una vez realizada la carga inicial, los datos aquí almacenados no se modificarán.

## **2.2. REQUISITOS FUNCIONALES**

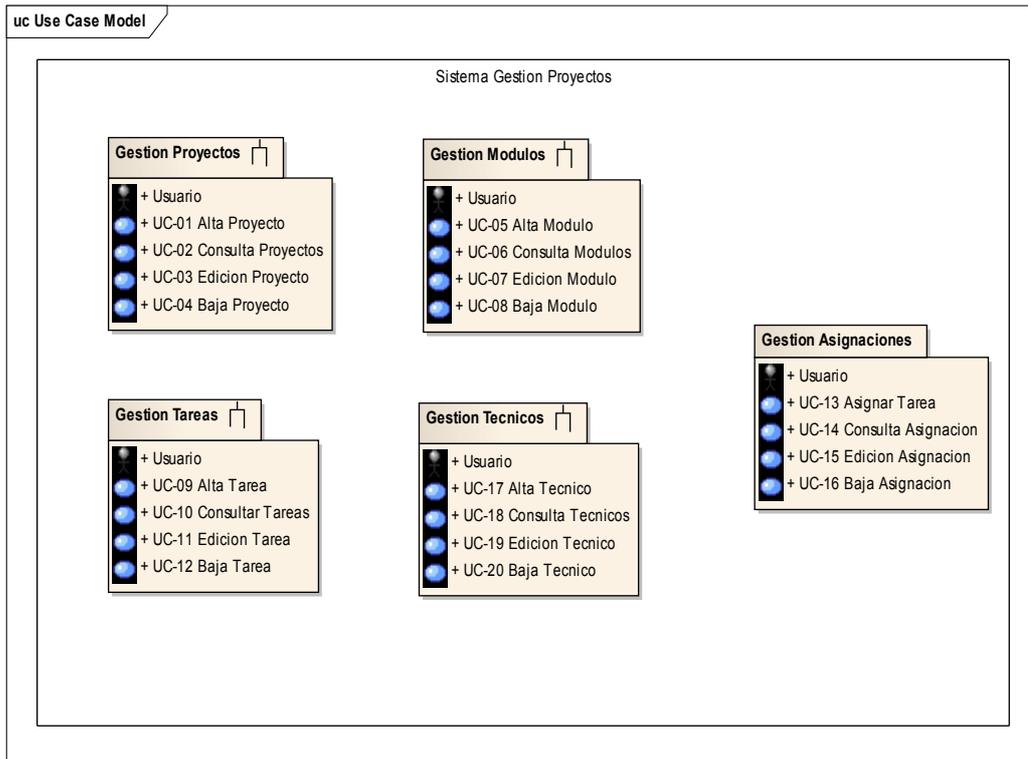


Figura: Diagrama de subsistemas

## 2.1.1. DIAGRAMAS DE CASOS DE USO

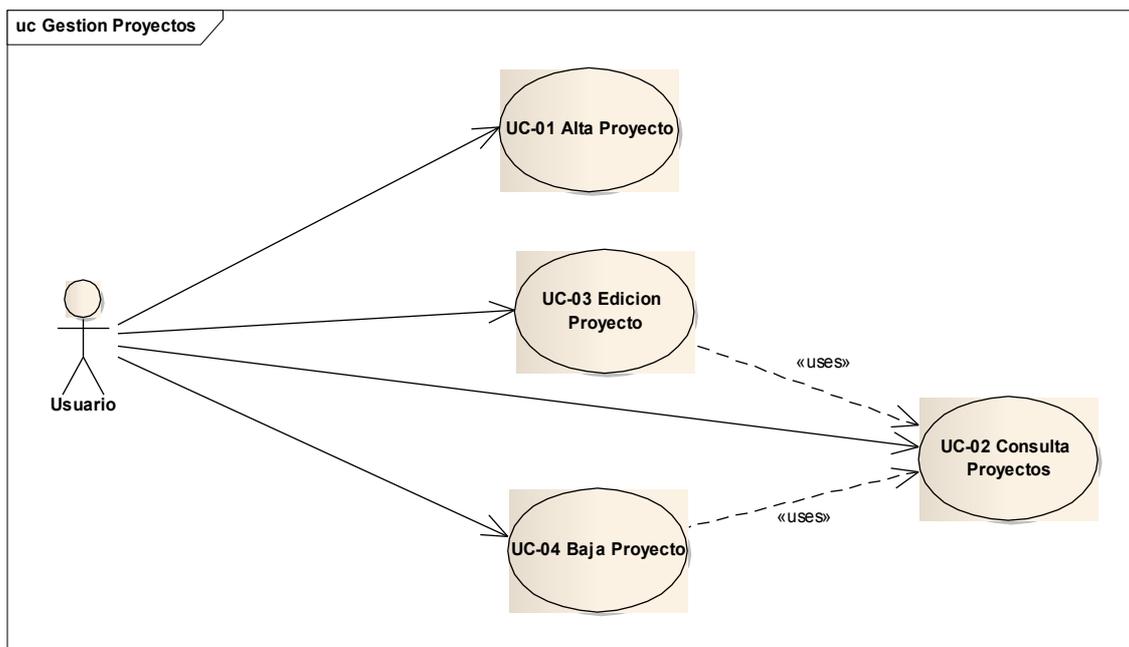
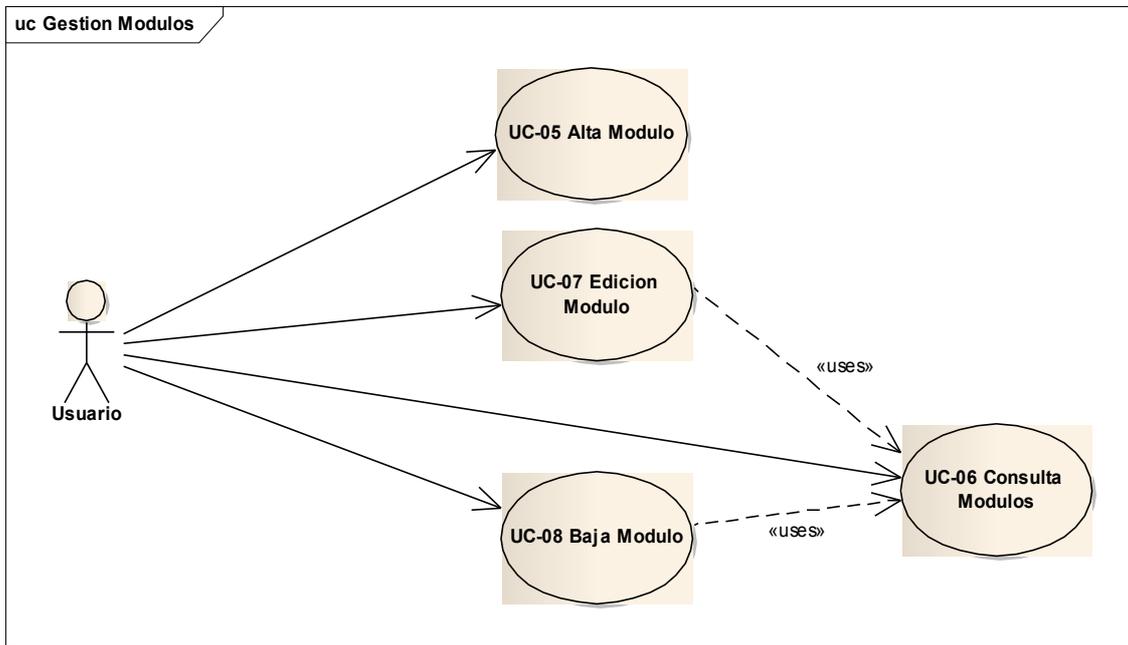
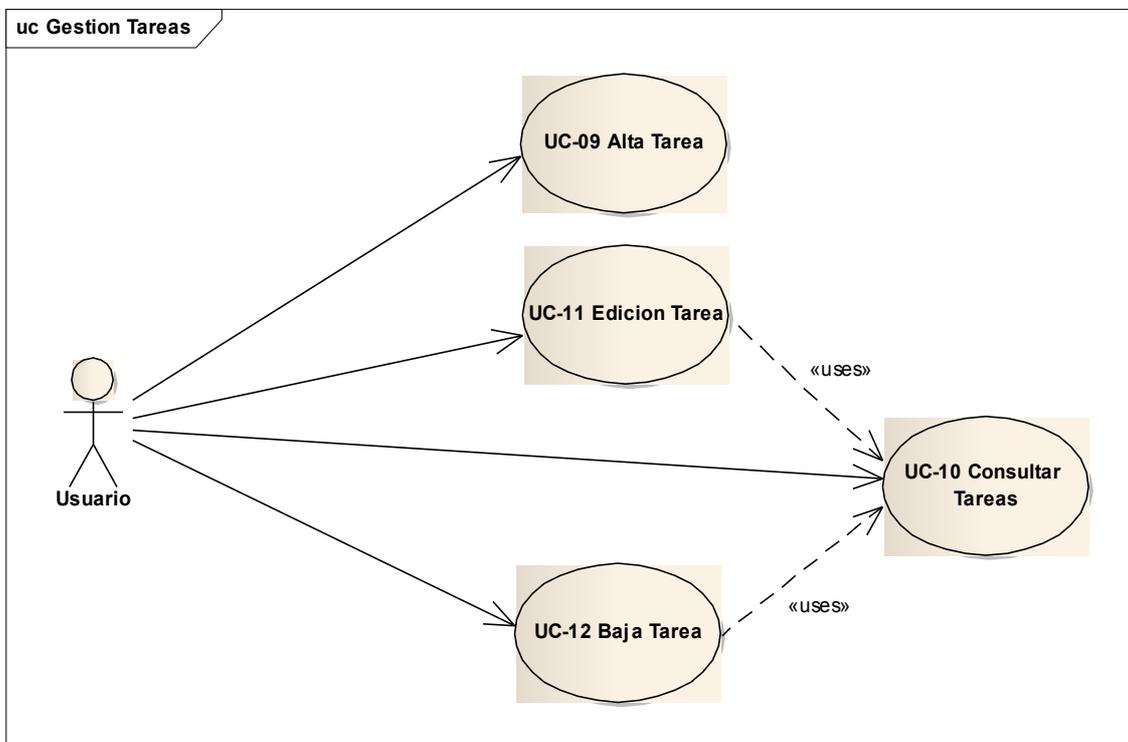


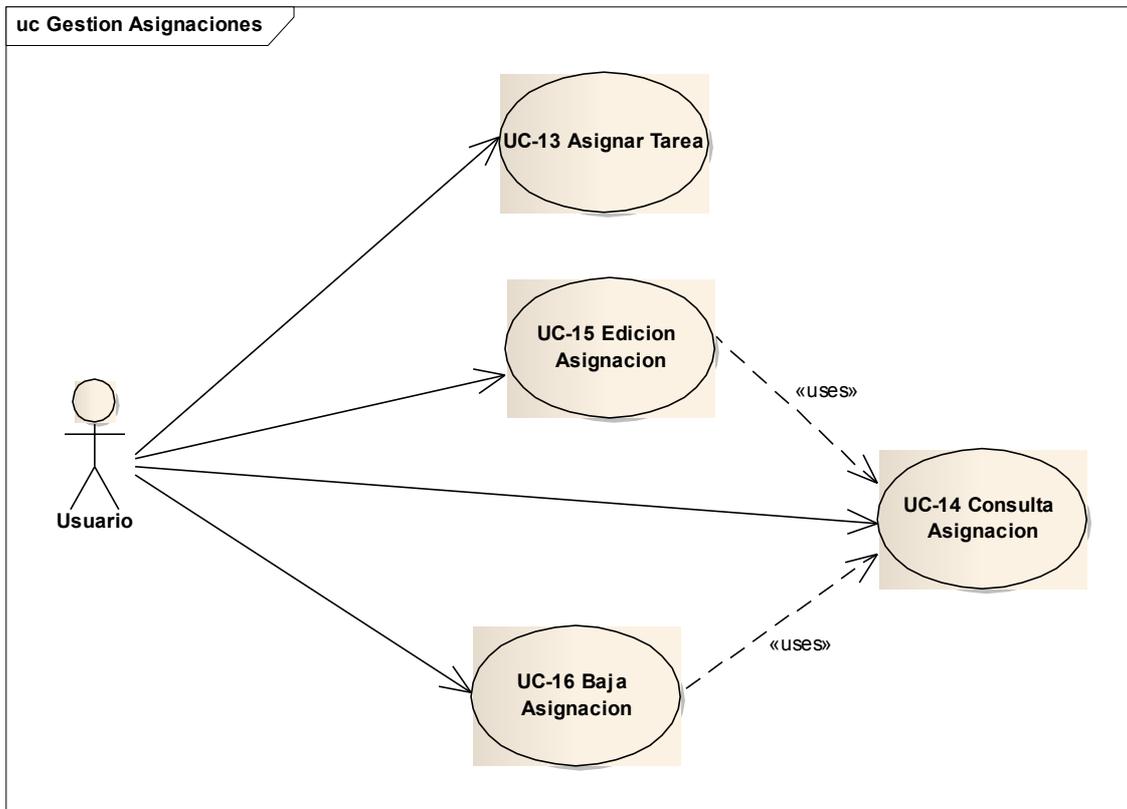
Figura: Diagrama de casos de uso del subsistema Gestión de Proyectos



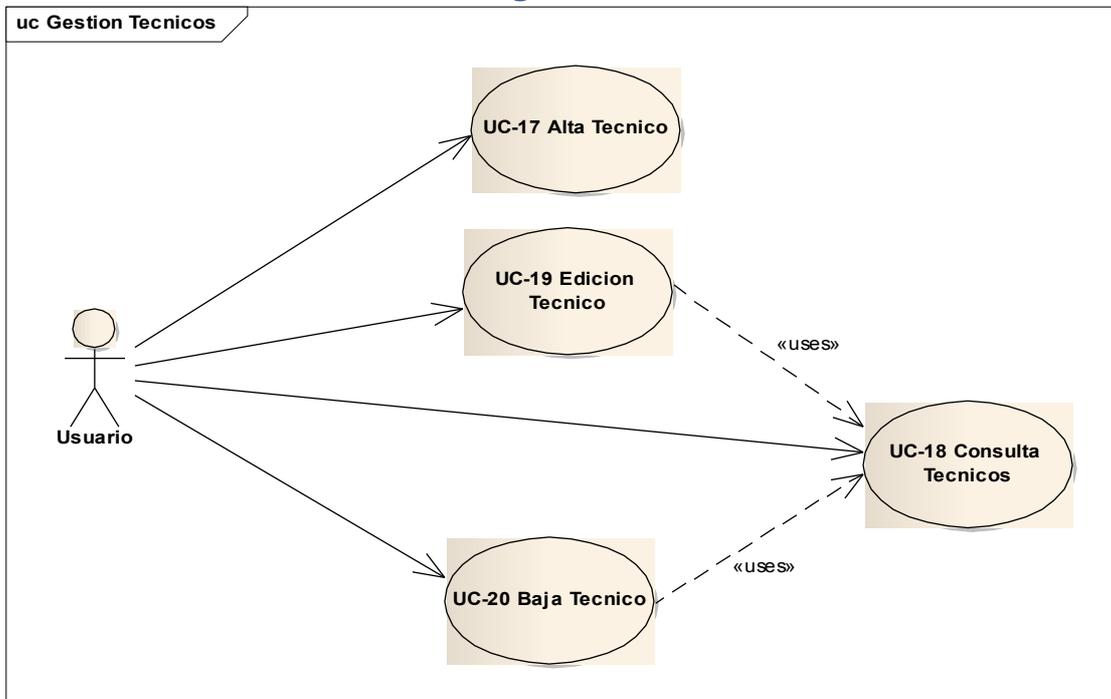
**Figura: Diagrama de casos de uso del subsistema Gestión de Módulos**



**Figura: Diagrama de casos de uso del subsistema Gestión de Tareas**



**Figura: Diagrama de casos de uso del subsistema Gestión de Asignaciones**



**Figura: Diagrama de casos de uso del subsistema Gestión de Técnicos**

### 2.2.2. DEFINICIÓN DE ACTORES

<b>ACT-01</b>	<b>Usuario</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Descripción</b>	Este actor representa al usuario de la aplicación
<b>Observaciones</b>	Será el encargado de llevar a cabo una gestión controlada de los proyectos vigentes en la organización, sus módulos, sus tareas y sus respectivas relaciones con los técnicos de la misma. Por tanto, será el actor origen y destino de la información recogida por el sistema

### 2.2.3. DESCRIPCIÓN TEXTUAL DE CASOS DE USO

<b>UC-01</b>	<b>Alta Proyecto</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Almacenar en el sistema un proyecto
<b>Descripción</b>	El actor inserta un nuevo proyecto con todos sus datos en el sistema
<b>Precondiciones</b>	El proyecto no está registrado en el sistema

<b>Postcondiciones</b>	El proyecto queda almacenado en el sistema
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de alta de un nuevo proyecto	
	2. El sistema solicita los datos del nuevo proyecto indicando cuales de ellos son obligatorios: nombre, coste y responsable del proyecto. Además, solicita otros datos de carácter opcional como: descripción, alcance, origen, fecha de inicio y fecha de finalización
3. El actor proporciona los datos requeridos por el sistema y solicita a éste que los almacene	
	4. El sistema almacena los datos introducidos por el actor, genera un identificador de proyecto que identifica unívocamente al nuevo proyecto registrado en el sistema e informa al administrador de que el proyecto ha sido registrado con éxito.  A continuación ejecuta el caso de uso Consulta de Proyectos
<b>Flujo Alternativo</b>	
3. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los datos introducidos por el actor.  Debe volverse al paso 3 del flujo normal para proseguir con este caso de uso	

<b>Excepciones</b>	
# 3. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor introduce una fecha de baja anterior a la fecha de alta del proyecto, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Proyectos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	La frecuencia de ejecución de este caso de uso será mucho más elevada durante los primeros meses de utilización

<b>UC-02</b>	<b>Consulta Proyectos</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Consultar la información de un proyecto
<b>Descripción</b>	El actor realiza una búsqueda con o sin filtros (en cuyo caso se mostrarían todos) y el sistema muestra la

	información del proyecto
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Se muestra la información del proyecto
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de proyectos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los proyectos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los proyectos registrados en el sistema. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	

<b>Excepciones</b>	
# 3. Si no hay proyectos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-03</b>	<b>Edición Proyecto</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Modificar los datos de un proyecto
<b>Descripción</b>	El actor modifica los datos de un proyecto existente en el sistema
<b>Precondiciones</b>	El proyecto está registrado en el sistema
<b>Postcondiciones</b>	Se almacenan en el sistema las modificaciones del proyecto, el cual sigue registrado en el sistema
<b>Incluye</b>	UC-02 Consulta Proyectos
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de proyectos	

2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los proyectos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
3. El actor solicita al sistema comenzar con el proceso de edición de proyectos, seleccionando el proyecto que desea modificar de la lista al pulsar en el enlace <i>Editar</i>	
	4. El sistema abre un popup con los siguientes datos correspondientes al proyecto a modificar: nombre, descripción, alcance, coste, responsable, origen, fecha de inicio y fecha de fin
	5. El sistema permite al actor modificar todos los campos del proyecto
6. El actor modifica los datos deseados y solicita al sistema que los almacene	
	7. El sistema modifica los datos del proyecto y cierra el popup de edición. A continuación, ejecuta el caso de uso Consulta de Proyectos
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los proyectos registrados en el sistema. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el	

sistema borra todos los datos introducidos por el actor.  Debe volverse al paso 6 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay proyectos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 6. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación este caso de uso continua	
# 6. Si el actor introduce una fecha de baja anterior a la fecha de alta del proyecto, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 6. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Proyectos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-04</b>	<b>Baja Proyecto</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Eliminar del sistema un proyecto

<b>Descripción</b>	El actor da de baja un proyecto en el sistema, borrando todos sus datos
<b>Precondiciones</b>	El proyecto está registrado en el sistema
<b>Postcondiciones</b>	El proyecto es eliminado del sistema
<b>Incluye</b>	UC-02 Consulta Proyectos
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de proyectos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los proyectos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
3. El actor solicita al sistema comenzar con el proceso de baja de proyectos, seleccionando el proyecto que desea eliminar de la lista al pulsar en el enlace <i>Eliminar</i>	
	4. El sistema solicita confirmación del borrado del proyecto
5. El actor confirma el borrado del proyecto	
	6. El sistema elimina el proyecto e informa al actor de que el proyecto ya no existe en el sistema. A continuación se ejecuta el caso de uso

	Consulta de Proyectos
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los proyectos registrados en el sistema. La información que se muestra para los proyectos es: nombre, descripción, alcance, origen, coste, responsable, fecha de inicio y fecha de finalización.
<b>Excepciones</b>	
# 3. Si no hay proyectos registrados, el sistema informa de esta situación.  A continuación, este caso de uso queda sin efecto	
# 5. Si el actor no confirma la operación de borrado del proyecto seleccionado, el sistema cancela el procedimiento. A continuación, este caso de uso queda sin efecto	
# 6. Si el proyecto a eliminar tiene dependencias en la base de datos del sistema, el sistema lanza una excepción solicitando que se eliminen en primer lugar sus dependencias. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	1 vez por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-05</b>	<b>Alta Módulo</b>
<b>Autor</b>	Reyes Martínez Durbán

<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Almacenar en el sistema un módulo
<b>Descripción</b>	El actor inserta un nuevo módulo con todos sus datos en el sistema
<b>Precondiciones</b>	El módulo no está registrado en el sistema
<b>Postcondiciones</b>	El módulo queda almacenado en el sistema
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de alta de un nuevo módulo	
	2. El sistema solicita los datos del nuevo módulo indicando cuales de ellos son obligatorios: nombre y proyecto al que pertenece. Además, solicita otros datos de carácter opcional como: descripción, fecha de inicio, fecha de finalización y observaciones
3. El actor proporciona los datos requeridos por el sistema y solicita a éste que los almacene	
	4. El sistema almacena los datos introducidos por el actor, genera un identificador de módulo que identifica unívocamente al nuevo módulo registrado en el sistema e informa al administrador de que el módulo ha sido registrado con éxito.  A continuación ejecuta el caso de uso

	Consulta de Módulos
<b>Flujo Alternativo</b>	
<p>3. Si el actor selecciona <i>limpiar</i>, el sistema borra todos los datos introducidos por el actor.</p> <p>Debe volverse al paso 3 del flujo normal para proseguir con este caso de uso</p>	
<b>Excepciones</b>	
# 3. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor introduce una fecha de baja anterior a la fecha de alta del módulo, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Módulos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	7-8 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	La frecuencia de ejecución de este caso de uso será mucho más elevada durante los primeros meses de utilización. La frecuencia de este caso de uso está ligada a la frecuencia de uso del caso UC-01 Alta Proyecto

<b>UC-06</b>	<b>Consulta Módulos</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Consultar la información de un módulo
<b>Descripción</b>	El actor realiza una búsqueda con o sin filtros (en cuyo caso se mostrarían todos) y el sistema muestra la información del módulo
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Se muestra la información del módulo
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de módulos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los módulos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los módulos registrados en el sistema. La información que se muestra

	para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay módulos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-07</b>	<b>Edición Módulo</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Modificar los datos de un módulo
<b>Descripción</b>	El actor modifica los datos de un módulo existente en el sistema
<b>Precondiciones</b>	El módulo está registrado en el sistema
<b>Postcondiciones</b>	Se almacenan en el sistema las modificaciones del módulo, el cual

	sigue registrado en el sistema
<b>Incluye</b>	UC-06 Consulta Módulos
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de consulta de módulos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los módulos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
3. El actor solicita al sistema comenzar con el proceso de edición de módulos, seleccionando el módulo que desea modificar de la lista al pulsar en el enlace <i>Editar</i>	
	4. El sistema abre un popup con los siguientes datos correspondientes al módulo a modificar: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
	5. El sistema permite al actor modificar todos los campos del módulo
6. El actor modifica los datos deseados y solicita al sistema que los almacene	
	7. El sistema modifica los datos del módulo y cierra el popup de edición. A continuación, ejecuta el caso de uso Consulta de Módulos
Flujo Alternativo	

2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los módulos registrados en el sistema. La información que se muestra para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los datos introducidos por el actor.  Debe volverse al paso 6 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay módulos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 6. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación este caso de uso continua	
# 6. Si el actor introduce una fecha de baja anterior a la fecha de alta del módulo, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 6. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Módulos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes

<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-08</b>	<b>Baja Módulo</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Eliminar del sistema un módulo
<b>Descripción</b>	El actor da de baja un módulo en el sistema, borrando todos sus datos
<b>Precondiciones</b>	El módulo está registrado en el sistema
<b>Postcondiciones</b>	El módulo es eliminado del sistema
<b>Incluye</b>	UC-06 Consulta Módulos
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de módulos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los módulos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
3. El actor solicita al sistema comenzar con el proceso de baja de módulos, seleccionando el módulo que desea	

eliminar de la lista al pulsar en el enlace <i>Eliminar</i>	
	4. El sistema solicita confirmación del borrado del módulo
5. El actor confirma el borrado del módulo	
	6. El sistema elimina el módulo e informa al actor de que el módulo ya no existe en el sistema. A continuación se ejecuta el caso de uso Consulta de Módulos
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los módulos registrados en el sistema. La información que se muestra para los módulos es: nombre, descripción, proyecto al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Excepciones</b>	
# 3. Si no hay módulos registrados, el sistema informa de esta situación.  A continuación, este caso de uso queda sin efecto	
# 5. Si el actor no confirma la operación de borrado del módulo seleccionado, el sistema cancela el procedimiento. A continuación, este caso de uso queda sin efecto	
# 6. Si el módulo a eliminar tiene dependencias en la base de datos del sistema, el sistema lanza una excepción solicitando que se eliminen en primer lugar sus dependencias. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta

<b>Frecuencia de uso</b>	1-2 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-
<b>UC-09</b>	<b>Alta Tarea</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Almacenar en el sistema una tarea
<b>Descripción</b>	El actor inserta una nueva tarea con todos sus datos en el sistema
<b>Precondiciones</b>	La tarea no está registrada en el sistema
<b>Postcondiciones</b>	La tarea queda almacenada en el sistema
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de alta de una nueva tarea	
	2. El sistema solicita los datos de la nueva tarea indicando cuales de ellos son obligatorios: nombre, proyecto y módulo al que pertenece. Además, solicita otros datos de carácter opcional como: descripción, fecha de inicio, fecha de finalización y observaciones
3. El actor proporciona los datos requeridos por el	

sistema y solicita a éste que los almacene	
	<p>4. El sistema almacena los datos introducidos por el actor, genera un identificador de tarea que identifica unívocamente a la nueva tarea registrada en el sistema e informa al administrador de que la tarea ha sido registrada con éxito.</p> <p>A continuación ejecuta el caso de uso Consulta de Tareas</p>
<b>Flujo Alternativo</b>	
<p>3. Si el actor selecciona <i>limpiar</i>, el sistema borra todos los datos introducidos por el actor.</p> <p>Debe volverse al paso 3 del flujo normal para proseguir con este caso de uso</p>	
<b>Excepciones</b>	
# 3. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor introduce una fecha de baja anterior a la fecha de alta del módulo, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 3. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Tareas	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	10-12 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	La frecuencia de ejecución de este caso de uso será mucho más elevada

	durante los primeros meses de utilización. La frecuencia de este caso de uso está ligada a la frecuencia de uso del caso UC-05 Alta Modulo
--	--

UC-10	Consulta Tareas
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Consultar la información de una tarea
<b>Descripción</b>	El actor realiza una búsqueda con o sin filtros (en cuyo caso se mostrarían todas) y el sistema muestra la información de la tarea
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Se muestra la información de la tarea
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de consulta de tareas	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con las tareas registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra para las tareas es: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones

<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todas las tareas registradas en el sistema. La información que se muestra para las tareas es: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay tareas registradas, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-11</b>	<b>Edición Tarea</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Modificar los datos de una tarea

<b>Descripción</b>	El actor modifica los datos de una tarea existente en el sistema
<b>Precondiciones</b>	La tarea está registrada en el sistema
<b>Postcondiciones</b>	Se almacenan en el sistema las modificaciones de la tarea, la cual sigue registrada en el sistema
<b>Incluye</b>	UC-10 Consulta Tareas
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de consulta de tareas	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con las tareas registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra para las tareas es: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
3. El actor solicita al sistema comenzar con el proceso de edición de tareas, seleccionando la tarea que desea modificar de la lista al pulsar en el enlace <i>Editar</i>	
	4. El sistema abre un popup con los siguientes datos correspondientes a la tarea a modificar: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
	5. El sistema permite al actor modificar todos los campos de la tarea
6. El actor modifica los datos deseados y solicita al sistema que los almacene	

	7. El sistema modifica los datos de la tarea y cierra el popup de edición. A continuación, ejecuta el caso de uso Consulta de Tareas
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todas las tareas registradas en el sistema La información que se muestra para las tareas es: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los datos introducidos por el actor.  Debe volverse al paso 6 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay tareas registradas, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 6. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación este caso de uso continua	
# 6. Si el actor introduce una fecha de baja anterior a la fecha de alta del módulo, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 6. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema	

ejecuta el caso de uso Consulta de Tareas	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	4-5 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

UC-12	Baja Tarea
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Eliminar del sistema una tarea
<b>Descripción</b>	El actor da de baja una tarea en el sistema, borrando todos sus datos
<b>Precondiciones</b>	La tarea está registrada en el sistema
<b>Postcondiciones</b>	La tarea es eliminada del sistema
<b>Incluye</b>	UC-10 Consulta Tareas
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de consulta de tareas	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con las tareas registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para las tareas es: nombre, descripción,

	proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
3. El actor solicita al sistema comenzar con el proceso de baja de tareas, seleccionando la tarea que desea eliminar de la lista al pulsar en el enlace <i>Eliminar</i>	
	4. El sistema solicita confirmación del borrado de la tarea
5. El actor confirma el borrado de la tarea	
	6. El sistema elimina la tarea e informa al actor de que la tarea ya no existe en el sistema. A continuación se ejecuta el caso de uso Consulta de Tareas
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todas las tareas registradas en el sistema. La información que se muestra para las tareas es: nombre, descripción, proyecto y módulo al que pertenece, fecha de inicio, fecha de finalización y observaciones
<b>Excepciones</b>	
# 3. Si no hay tareas registradas, el sistema informa de esta situación.  A continuación, este caso de uso queda sin efecto	
# 5. Si el actor no confirma la operación de borrado de la tarea seleccionada, el sistema cancela el procedimiento. A continuación, este caso de uso queda sin efecto	
# 6. Si la tarea a eliminar tiene dependencias en la base de datos	

del sistema, el sistema lanza una excepción solicitando que se eliminen en primer lugar sus dependencias. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-13</b>	<b>Asignar Tarea</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Asignar una tarea a un técnico
<b>Descripción</b>	El actor realiza una búsqueda de tareas y técnicos registrados en el sistema y asigna la tarea al técnico
<b>Precondiciones</b>	La tarea y el técnico están registrados en el sistema
<b>Postcondiciones</b>	La asignación tarea-técnico queda almacenada en el sistema
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de asignación de tareas	
2. El actor selecciona el proyecto	
	3. El sistema carga el combo de módulos con los módulos del proyecto

	seleccionado
4. El actor selecciona el módulo	
	5. El sistema carga el combo de tareas con las tareas del módulo seleccionado
6. El actor selecciona la tarea	
7. El actor selecciona el técnico	
8. El actor solicita al sistema que almacene la asignación	
	9. El sistema almacena la asignación, genera un identificador de asignación que identifica unívocamente a la nueva asignación registrada en el sistema e informa al administrador de que la asignación ha sido registrada con éxito. La nueva asignación se añade al listado de asignaciones
<b>Flujo Alternativo</b>	
2. El actor selecciona directamente la tarea	
3. El actor selecciona el técnico	
4. El actor solicita al sistema que almacene la asignación	
	5. El sistema almacena la asignación, genera un identificador de asignación que identifica unívocamente a la nueva asignación registrada en el sistema e informa al administrador de que la asignación ha sido registrada con éxito. La nueva asignación se añade al listado de asignaciones
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	

<b>Excepciones</b>	
# 6. Si no hay tareas registradas, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 7. Si no hay tareas técnicos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 8. Si el actor no selecciona una tarea y un técnico el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación este caso de uso continua	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-14</b>	<b>Consulta Asignación</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Consultar los datos de una asignación de una tarea a un técnico
<b>Descripción</b>	El actor realiza una búsqueda con o sin filtros (en cuyo caso se mostrarían todas) y el sistema muestra la información de las asignaciones de una tarea a un técnico
<b>Precondiciones</b>	-

<b>Postcondiciones</b>	Se muestra la información de la asignación
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de asignaciones	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con las asignaciones registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todas las asignaciones registradas en el sistema. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	

# 3. Si no hay tareas registradas, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

UC-15	Edición Asignación
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Modificar los datos de una asignación de una tarea a un técnico
<b>Descripción</b>	El actor modifica los datos de una asignación de una tarea a un técnico
<b>Precondiciones</b>	La tarea y el técnico están registrados en el sistema
<b>Postcondiciones</b>	Se almacenan en el sistema las modificaciones de la asignación tarea-técnico.
<b>Incluye</b>	UC-14 Consulta Asignación
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de asignación de tareas	

<p>2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i></p>	
	<p>3. El sistema muestra un listado con las asignaciones registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización</p>
<p>3. El actor solicita al sistema comenzar con el proceso de edición de asignación, seleccionando la asignación que desea modificar de la lista al pulsar en el enlace <i>Editar</i></p>	
	<p>4. El sistema abre un popup con los siguientes datos correspondientes a la asignación a modificar: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización</p>
	<p>5. El sistema permite al actor modificar fecha de inicio y fecha de finalización</p>
<p>6. El actor modifica los datos deseados y solicita al sistema que los almacene</p>	
	<p>7. El sistema modifica los datos de la asignación y cierra el popup de edición. A continuación, ejecuta el caso de uso Consulta de Asignación</p>
<b>Flujo Alternativo</b>	
<p>2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i></p>	
	<p>3. El sistema muestra un listado con todas las asignaciones registradas en el sistema. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización</p>
<b>Flujo Alternativo</b>	
<p>2. Si el actor selecciona <i>limpiar</i>, el</p>	

sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 2. Si no hay tareas o técnicos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 6. Si el actor introduce una fecha de baja anterior a la fecha de alta de la asignación, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
<b>Prioridad</b>	
<b>Frecuencia de uso</b>	3-4 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-16</b>	<b>Baja Asignación</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Eliminar del sistema una asignación de una tarea a un técnico
<b>Descripción</b>	El actor da de baja una asignación de una tarea a un técnico en el sistema, borrando todos sus datos
<b>Precondiciones</b>	La tarea y el técnico están registrados

	en el sistema
<b>Postcondiciones</b>	La asignación tarea-técnico es eliminada del sistema
<b>Incluye</b>	UC-14 Consulta Asignación
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de asignación de tareas	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con las asignaciones registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización
3. El actor solicita al sistema comenzar con el proceso de eliminación de asignación, seleccionando la asignación que desea modificar de la lista al pulsar en el enlace <i>Eliminar</i>	
	4. El sistema solicita confirmación del borrado de la asignación
5. El actor confirma el borrado de la asignación	
	6. El sistema elimina la asignación e informa al actor de que la asignación ya no existe en el sistema. A continuación se ejecuta el caso de uso Consulta de Asignación
<b>Flujo Alternativo</b>	

2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todas las asignaciones registradas en el sistema que cumplen los criterios de búsqueda. La información que se muestra es: proyecto, módulo, tarea, técnico, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 2. Si no hay tareas o técnicos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 5. Si el actor no confirma la operación de borrado del técnico seleccionado, el sistema cancela el procedimiento. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	
<b>Frecuencia de uso</b>	3-4 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-17</b>	<b>Alta Técnico</b>
<b>Autor</b>	Reyes Martínez Durbán

<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Almacenar en el sistema un técnico
<b>Descripción</b>	El actor inserta un nuevo técnico con todos sus datos en el sistema
<b>Precondiciones</b>	El técnico no está registrado en el sistema
<b>Postcondiciones</b>	El técnico queda almacenado en el sistema
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
Flujo Normal	
Intenciones de usuario	Obligaciones de sistema
1. El actor solicita al sistema comenzar con el proceso de alta de un nuevo técnico	
	2. El sistema solicita los datos del nuevo técnico indicando cuales de ellos son obligatorios: nombre, apellidos, NIF y perfil. Además, solicita otros datos de carácter opcional como: fecha de inicio y fecha de finalización
3. El actor proporciona los datos requeridos por el sistema y solicita a éste que los almacene	
	4. El sistema almacena los datos introducidos por el actor, genera un identificador de técnico que identifica unívocamente al nuevo técnico registrado en el sistema e informa al administrador de que el técnico ha sido registrado con éxito.

	A continuación ejecuta el caso de uso Consulta de Técnicos
<b>Flujo Alternativo</b>	
<p>3. Si el actor selecciona <i>limpiar</i>, el sistema borra todos los datos introducidos por el actor.</p> <p>Debe volverse al paso 3 del flujo normal para proseguir con este caso de uso</p>	
<b>Excepciones</b>	
<p># 3. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua</p>	
<p># 3. Si el actor introduce una fecha de baja anterior a la fecha de alta del proyecto, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua</p>	
<p># 3. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Técnicos</p>	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	La frecuencia de ejecución de este caso de uso será mucho más elevada durante los primeros meses de utilización

<b>UC-18</b>	<b>Consulta Técnicos</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Consultar la información de un técnico
<b>Descripción</b>	El actor realiza una búsqueda con o sin filtros (en cuyo caso se mostrarían todos) y el sistema muestra la información del técnico
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Se muestra la información del técnico
<b>Incluye</b>	-
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de técnicos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los técnicos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los técnicos registrados

	en el sistema. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los filtros introducidos por el actor.  Debe volverse al paso 2 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay técnicos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	15-20 veces por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-19</b>	<b>Edición Técnico</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Modificar los datos de un técnico
<b>Descripción</b>	El actor modifica los datos de un técnico existente en el sistema
<b>Precondiciones</b>	El técnico está registrado en el sistema
<b>Postcondiciones</b>	Se almacenan en el sistema las modificaciones del técnico, el cual sigue registrado en el sistema
<b>Incluye</b>	UC-18 Consulta Técnicos

<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de técnicos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los técnicos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización
3. El actor solicita al sistema comenzar con el proceso de edición de técnicos, seleccionando el técnico que desea modificar de la lista al pulsar en el enlace <i>Editar</i>	
	4. El sistema abre un popup con los siguientes datos correspondientes al técnico a modificar: nombre, apellidos, NIF, perfil, fecha de inicio y fecha de fin.
	5. El sistema permite al actor modificar todos los campos del técnico salvo el NIF
6. El actor modifica los datos deseados y solicita al sistema que los almacene	
	7. El sistema modifica los datos del técnico y cierra el popup de edición. A continuación, ejecuta el caso de uso Consulta de Módulos
<b>Flujo Alternativo</b>	

2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los técnicos registrados en el sistema. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización
<b>Flujo Alternativo</b>	
2. Si el actor selecciona <i>limpiar</i> , el sistema borra todos los datos introducidos por el actor.  Debe volverse al paso 6 del flujo normal para proseguir con este caso de uso	
<b>Excepciones</b>	
# 3. Si no hay técnicos registrados, el sistema informa de esta situación. A continuación, este caso de uso queda sin efecto	
# 6. Si el actor no proporciona alguno de los datos obligatorios, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación este caso de uso continua	
# 6. Si el actor introduce una fecha de baja anterior a la fecha de alta del técnico, el sistema informa de esta situación y solicita nuevamente la entrada de estos datos. A continuación, este caso de uso continua	
# 6. Si el actor solicita <i>cancelar</i> la operación, este caso de uso queda sin efecto y a continuación, el sistema ejecuta el caso de uso Consulta de Técnicos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	2-3 veces por mes

<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

<b>UC-20</b>	<b>Baja Técnico</b>
<b>Autor</b>	Reyes Martínez Durbán
<b>Versión</b>	1.0 (05/04/2011)
<b>Actores</b>	Usuario
<b>Propósito</b>	Eliminar del sistema un técnico
<b>Descripción</b>	El actor da de baja un técnico en el sistema, borrando todos sus datos
<b>Precondiciones</b>	El técnico está registrado en el sistema
<b>Postcondiciones</b>	El técnico es eliminado del sistema
<b>Incluye</b>	UC-19 Consulta Técnicos
<b>Extiende</b>	-
<b>Hereda de</b>	-
<b>Flujo Normal</b>	
<b>Intenciones de usuario</b>	<b>Obligaciones de sistema</b>
1. El actor solicita al sistema comenzar con el proceso de consulta de técnicos	
2. El actor introduce algún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con los técnicos registrados en el sistema que cumplen los criterios de búsqueda. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización

3. El actor solicita al sistema comenzar con el proceso de baja de técnicos, seleccionando el técnico que desea eliminar de la lista al pulsar en el enlace <i>Eliminar</i>	
	4. El sistema solicita confirmación del borrado del técnico
5. El actor confirma el borrado del técnico	
	6. El sistema elimina el técnico e informa al actor de que el técnico ya no existe en el sistema. A continuación se ejecuta el caso de uso Consulta de Técnicos
<b>Flujo Alternativo</b>	
2. El actor no introduce ningún criterio de búsqueda y pulsa <i>buscar</i>	
	3. El sistema muestra un listado con todos los técnicos registrados en el sistema. La información que se muestra para los técnicos es: nombre, apellidos, NIF, puesto, fecha de inicio y fecha de finalización
<b>Excepciones</b>	
# 3. Si no hay técnicos registrados, el sistema informa de esta situación.  A continuación, este caso de uso queda sin efecto	
# 5. Si el actor no confirma la operación de borrado del técnico seleccionado, el sistema cancela el procedimiento. A continuación, este caso de uso queda sin efecto	
# 6. Si el técnico a eliminar tiene dependencias en la base de datos del sistema, el sistema lanza una excepción solicitando que se eliminen en primer lugar sus dependencias. A continuación, este caso de uso queda	

sin efecto	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	1 vez por mes
<b>Requisitos no funcionales</b>	-
<b>Observaciones</b>	-

## 2.3. REQUISITOS NO FUNCIONALES

En este apartado se identifican los requisitos no funcionales, los cuales se corresponden normalmente con requisitos de carácter técnico o legal.

- Servidor de Bases de Datos

Gestor de BBDD MySQL, versión 5.5.20

- Entorno de Explotación

El sistema debe funcionar en un entorno de PC's que tengan acceso a Internet como mínimo con módem a 56K y utilicen como explorador Microsoft Internet Explorer versión 7 como mínimo, aunque debe ser también accesible en otros navegadores.

- Servidor Web

El sistema utilizará el Servidor de Tomcat, versión 6.0.35

## 3. APLICACIÓN

### 3.1. PROCESO DE INSTALACIÓN

Para compilar y ejecutar un programa java, es necesario instalar la plataforma java JDK (Java Development Kit). El JDK es el conjunto

básico de herramientas para compilar y ejecutar los programas java. La última versión de JDK se puede descargar gratuitamente de la página oficial de Sun.

Al ejecutar el instalador descargado, éste instalará ambos, el JDK (Java Development Kit) y el JRE (Java Runtime). Por defecto, el JDK y el JRE se instalarán en los directorios "C:\Archivos de programa\Java\jdk1.7.0\_02" y "C:\ Archivos de programa\Java\jre7", respectivamente.

Para tener disponible el compilador de java y el entorno de ejecución para compilar y ejecutar los programas java, además, se tienen que establecer las variables de entorno. A la variable Path de variables de sistema se le tiene que añadir la ruta donde se tiene instalado el jdk (por ejemplo: C:\Archivos de programa\Java\jdk1.7.0\_02\bin) y establecer el valor de la variable JAVA\_HOME (por ejemplo: C:\Archivos de programa\Java\jdk1.7.0\_02). Esta variable la usan otras aplicaciones, como por ejemplo, el servidor Tomcat, para encontrar el directorio de instalación del jdk.

Ahora habría que instalar un Entorno de desarrollo integrado (IDE) para hacer el trabajo más sencillo a la hora de hacer programas en java y manejarnos con un número alto de clases.

### **Instalación de un entorno de desarrollo integrado (IDE): Eclipse.**

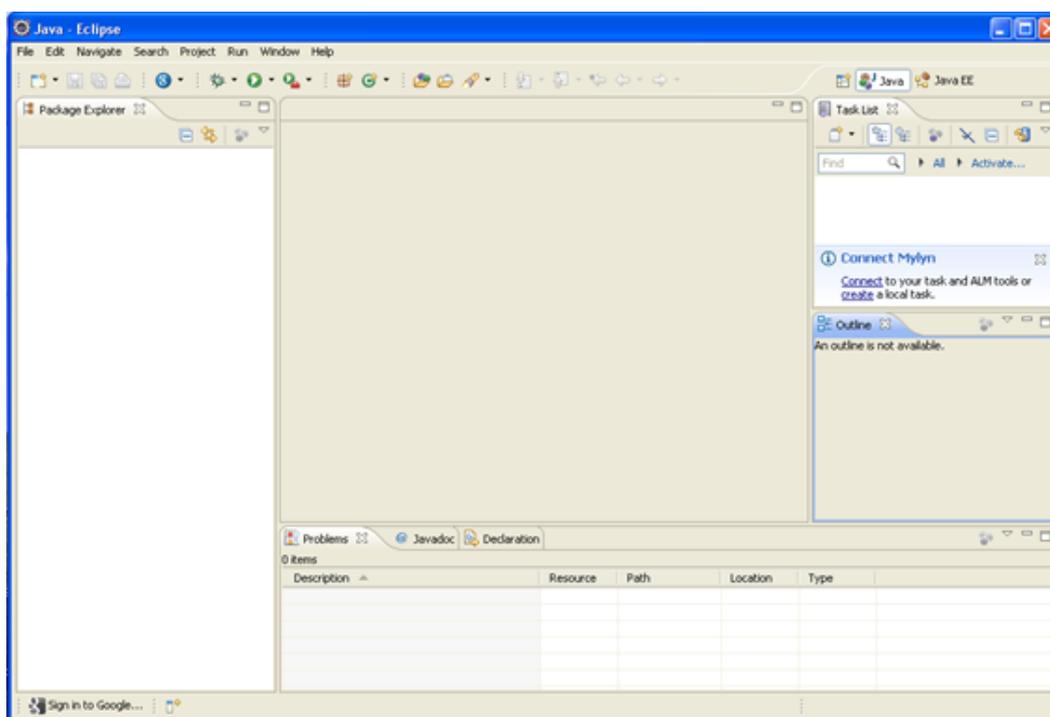
Un entorno de desarrollo integrado (IDE) puede hacer el trabajo mucho más sencillo, sobretodo si el desarrollo maneja un buen número de Clases. Además estos entornos nos permiten mucha más versatilidad para depurar programas puesto que tienen debuggers mucho más avanzados.

El IDE elegido fue Eclipse ya que tiene interesantes utilidades para Java, es gratuito, de código abierto y es relativamente sencillo de utilizar.

Su instalación es muy sencilla, se puede descargar de [www.eclipse.org](http://www.eclipse.org) en forma de archivo ZIP y solo hay que

descomprimirlo en la carpeta donde se quiera tenerlo instalado. En nuestro caso, la versión descargada fue la 3.7 Indigo.

Para ejecutarlo solo hay que arrancar el fichero Eclipse.exe. Una vez arrancado lo único que pedirá es la ruta por defecto donde eclipse guardará los proyectos que se creen.



Figura

### ra: Vista principal Eclipse

## Instalación de las herramientas y servidores necesarios para la gestión de la base de datos.

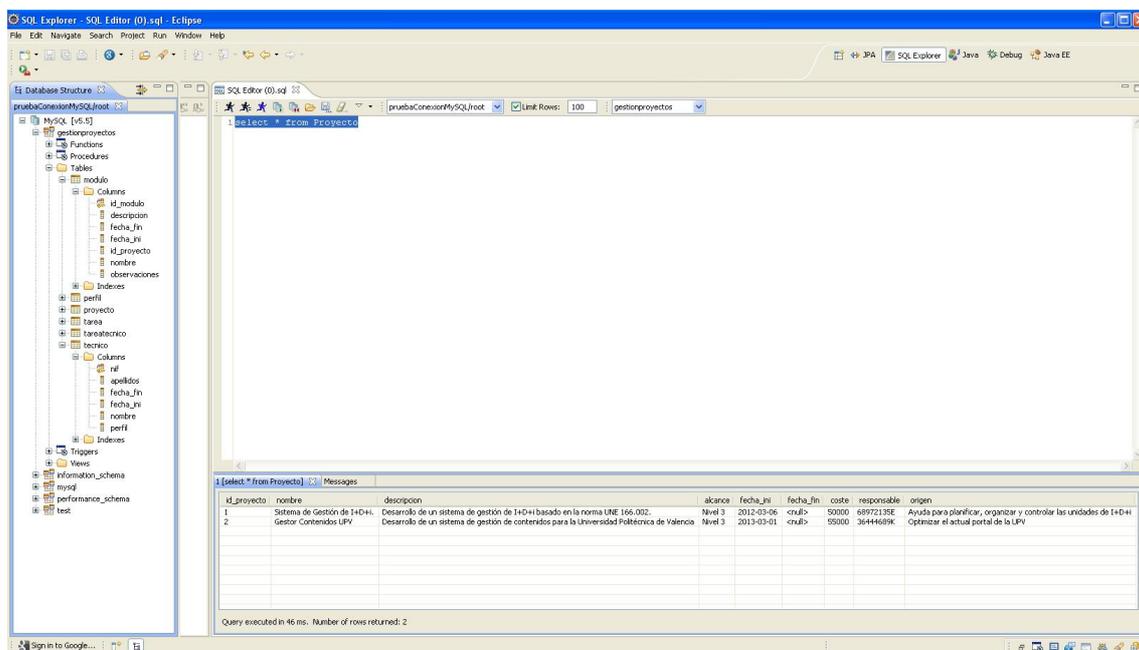
Para el desarrollo de la aplicación es necesaria una base de datos para gestionar los datos de la misma. La elegida fue MySQL, porque es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales. La versión utilizada es la 5.5.20 y se puede descargar de la web de MySQL. Por tanto, ésta será la utilizada para la gestión de la base de datos, tanto para su

creación, modificación de tablas, creación de claves ajenas, claves primarias, etc.

Para poder utilizar desde Eclipse el gestor de base de datos MySQL a través de JDBC, es necesario configurar lo que se conoce como el Conector/J para MySQL, es decir, una librería proporcionada por Oracle para el manejo de las funciones de acceso a datos con dicho motor. En la página de MySQL se puede encontrar la última versión del driver para su descarga, en este caso, la 5.1.18. Para utilizarlo, será suficiente con descomprimirlo en cualquier carpeta y establecer su ruta en el proyecto de eclipse para poderla

Si se quiere manipular la BBDD desde eclipse, se debe de instalar un plugin como Eclipse SQL Explorer. Se trata de un plugin de código fuente abierto, que permite la manipulación de bases de datos desde el propio Eclipse (ejecutar consultas SQL de todo tipo mostrando el resultado (SQL Results), estructura de la base de datos (Database structure), etc.

El aspecto de que mostraría desde eclipse sería:



**Figura: Plugin para Eclipse SQL Explorer**

Para acceder a MySQL desde Eclipse con SQL Explorer y el JDBC se debe de crear un nuevo profile para la conexión, en la ventana "Create New Connection Profile", con el nombre de la conexión y el driver que utiliza.

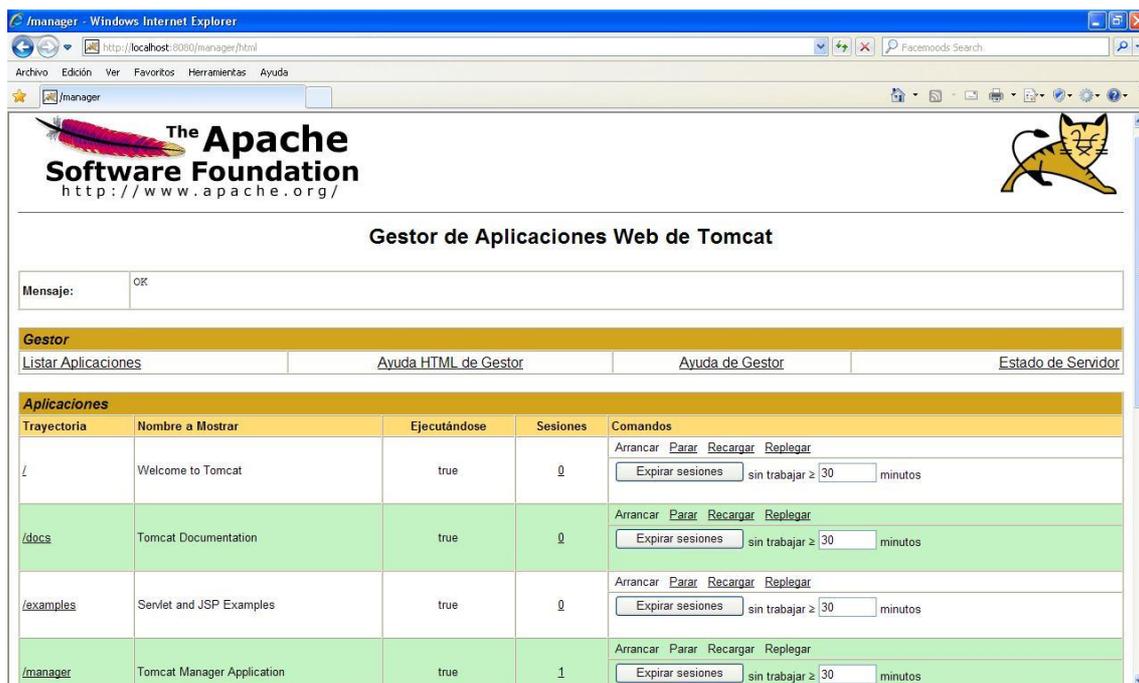
En la configuración del acceso a la base de datos, se debe de establecer la url de acceso, que sería el nombre o la ip del servidor de MySQL, en este caso correría bajo localhost en el puerto 3307. El nombre asignado a la base de datos creada fue gestionproyectos.

También es necesario un usuario para conectarse, en este caso root, y un password, que se correspondería con reyes.

A partir de aquí ya se puede utilizar Eclipse SQL Explorer para cualquier operación sobre la base de datos.

### **Instalación del servidor de aplicaciones Apache Tomcat**

Para el desarrollo de la aplicación se puede utilizar un servidor de aplicaciones, en este caso el elegido sería Apache Tomcat, en concreto la versión 6.0.35.



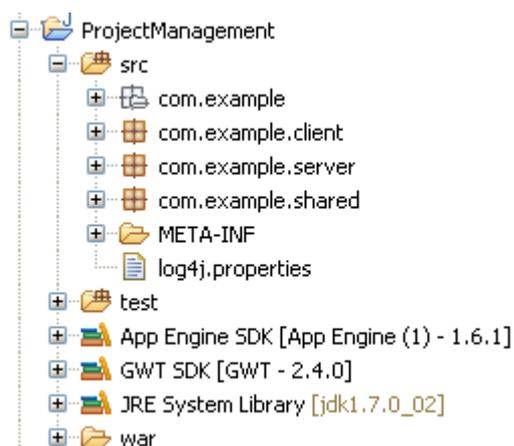
**Figura: Apache Tomcat Manager**

Una vez instalado eclipse y Tomcat en el equipo, y configurando las características pertinentes, si se quiere arrancar y parar continuamente el servidor Tomcat desde el entorno de eclipse, se tendrá que añadir el plugin del Tomcat a eclipse para que esto se pueda llevar a cabo. Para desplegar la aplicación en el servidor, sólo será necesario copiar el contenido de la carpeta war tras la compilación al directorio webapps del Tomcat.

Como la aplicación será desarrollada utilizando GWT, además de lo anterior, se debe de instalar el Google Web Toolkit Plugin para la versión de eclipse que tengamos.

La versión del plugin utilizada en este caso es la 2.4.0.

Una vez tengamos descargado e instalado el plugin de GWT, se debe de crear un nuevo proyecto mediante la opción “Web Application Project”, que nos creará un proyecto con la siguiente estructura:



## 3.2. BASE DE DATOS

Una Base de Datos es el conjunto de datos almacenados con una estructura lógica. Tan importante como los datos, es la estructura conceptual con la que se relacionan entre ellos.

En la práctica, se puede pensar en esto como el conjunto de datos más los programas (o software) que hacen de ellos un conjunto consistente.

Si no se tienen los dos factores unidos, no se puede hablar de una base de datos, ya que ambos combinados dan la coherencia necesaria para poder trabajar con los datos de una manera sistemática.

Al modelar una base de datos, se desea evitar puntos que crean confusión, duplicación de la información y por ende, un mal funcionamiento y exploración de la información.

Entre las propiedades indeseables en un diseño de bases de datos se encuentran:

- Redundancia en la información.
- Incapacidad de representar cierta información.

### **1. Primera forma normal**

Una relación está en primera forma normal (1FN) si y sólo si todos los dominios son atómicos. Un dominio es atómico si los elementos del dominio son indivisibles.

Es decir, no tenemos grupos de repetición o un conjunto de valores asociados repetidos asociados a una misma tupla.

### **2. Segunda forma normal**

Una relación está en segunda forma normal (2FN) si y sólo si está en 1FN y todos los atributos que no sean llaves dependen por completo de llave primaria.

### **3. Tercera forma normal**

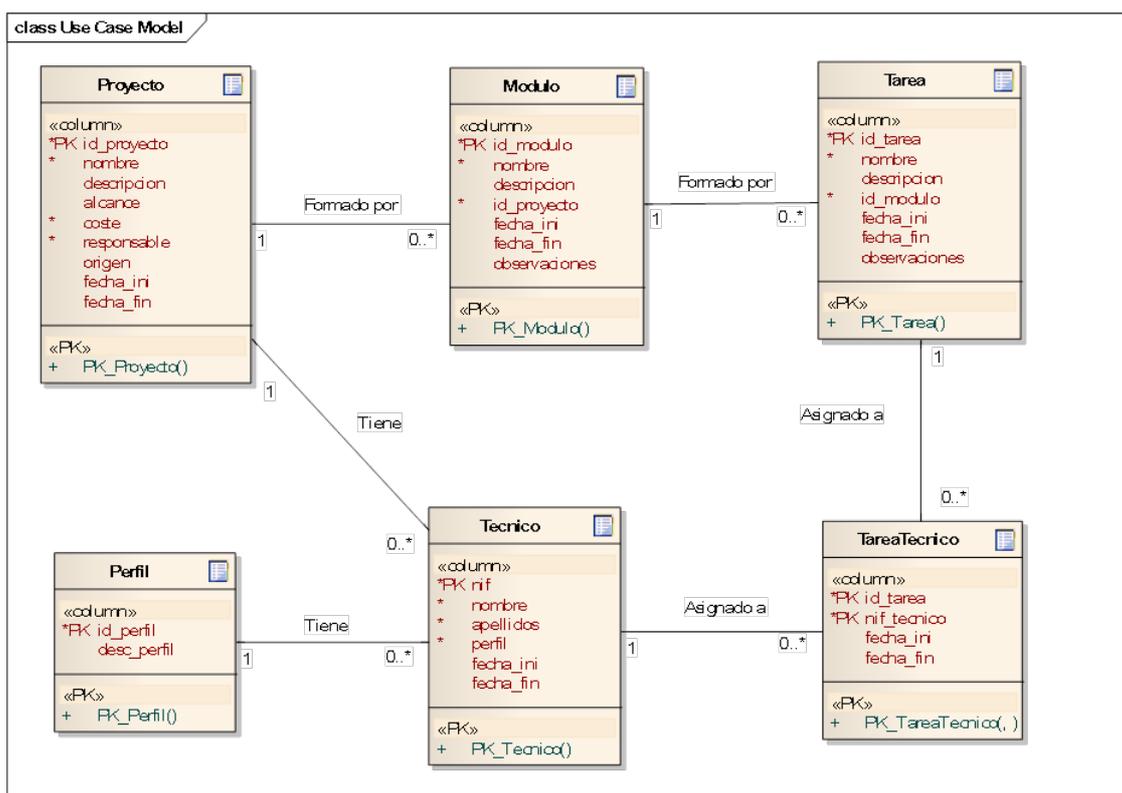
Una relación están en tercera forma normal (3FN) si y sólo si están en 2FN y todos los atributos no llave dependen de manera no transitiva de la llave primaria.

### **Integridad relacional**

En un momento dado, los valores de los datos en una base de datos son una representación de un fragmento de la realidad. Es decir, si tenemos una tabla con los atributos de personas y entre ellos el peso o la edad, estos no pueden ser negativos, porque en el mundo real, esto no es posible. Si añadimos una restricción de este tipo a una base de datos, estamos incluyéndole una regla de integridad.

Una vez comentadas algunos conceptos básicos sobre base de datos, para esta aplicación la base de datos se compone de 6 tablas: Proyecto, Modulo, Tarea, Tecnico, TareaTecnico y Perfil.

El modelo de datos para la aplicación sería:



**Figura: Modelo de Datos**

Aquellos atributos que empiezan con PK, son claves primarias, por lo que no pueden estar repetidos, deben tener un identificador único, para ello se han utilizan las secuencias, una por cada clave primaria, de manera que los valores de ese atributo es autoincrementable en uno. Utilizando MySQL esto se conseguiría utilizando la palabra “AUTO\_INCREMENT” en la declaración del atributo clave primaria de cada tabla.

### 3.3. DESCRIPCIÓN DE LA APLICACIÓN

Al acceder a la aplicación, se muestra una página en blanco con el menú disponible con las siguientes entradas: Proyecto, Módulo, Tarea y Técnico.

Al seleccionar cada una de ellas, la aplicación muestra las posibles opciones, que serían, Alta y Consulta, estando dentro de Consulta las opciones de Edición y Eliminación.

Dentro de la opción de Tarea, se encuentra el submenú “Asignar Tarea” que permite tanto asignar una tarea a un técnico como modificar o eliminar dicha asignación.

### Gestión de Proyectos:

La opción de alta de un nuevo proyecto lleva a un formulario donde se podrán introducir los campos asociados a dicho proyecto, en su caso, Nombre, Descripción, Alcance, que sería la prioridad del proyecto dentro de la organización, Coste total del proyecto, Técnico responsable, Origen o área en que se desarrolla el proyecto, Fecha de inicio, por defecto sería la fecha actual y Fecha de finalización del proyecto.

Para facilitar la tarea de asignar a un técnico responsable del proyecto, el sistema muestra una lista con todos los técnicos que pertenecen actualmente a la organización.

Los campos marcados con un \* serían obligatorios.



Formulario "Nuevo Proyecto" con los siguientes campos:

- Nombre: \*
- Descripción:
- Alcance:
- Coste: \*
- Responsable: \*
- Origen:
- Fecha inicio:
- Fecha fin:

Botones:

La opción de consulta de nuevo proyecto permite introducir diferentes criterios de búsqueda, mostrando por defecto un listado de todos los proyectos existentes en el sistema.

Existe la posibilidad de mostrar en el listado los proyectos dados de baja haciendo click en el check “Incluir proyectos dados de baja”.

**Consulta Proyectos**

**Criterios de búsqueda**

Nombre:

Descripcion:

Coste:

Responsable: Selecciona una opcion

Fecha Inicio:

Fecha Fin:

Incluir proyectos dados de baja

**Listado Proyectos**

Nombre	Descripcion	Alcance	Origen	Coste	Responsable	Fecha Alta	Fecha Baja		
Gestor Contenidos UPV	Desarrollo de un sistema de gestión de contenidos para la Universidad Politécnica de Valencia	Nivel 3	Optimizar el actual portal de la UPV	55000	Juan Garcia Gonzalez	01/03/2013		<a href="#">Editar</a>	<a href="#">Eliminar</a>
Sistema de Gestión de I+D+i.	Desarrollo de un sistema de gestión de I+D+i basado en la norma UNE 166.002.	Nivel 3	Ayuda para planificar, organizar y controlar las unidades de I+D+i	50000	Gema Martínez Gimeno	06/03/2012		<a href="#">Editar</a>	<a href="#">Eliminar</a>

1-2 of 2

Para cada proyecto del listado, se muestran dos links, Editar y Eliminar.

Si se hace click sobre Editar, el sistema abre una ventana de pop up con un formulario relleno con los datos del proyecto.

El usuario tiene la opción de modificar cualquier campo del proyecto y pulsar sobre Guardar si desea guardar sus cambios o Cancelar para desecharlos. En ambos casos, la aplicación cierra el pop up y vuelve al listado de proyectos anterior, actualizándolo.

Si el usuario hace click sobre Eliminar, el sistema muestra un mensaje de confirmación de eliminación del proyecto en primera plana, con las opciones ok y cancel. Si el usuario selecciona ok, el proyecto es eliminado del sistema y se vuelve al listado de proyectos, habiéndose suprimido de éste el proyecto seleccionado. Si el usuario selecciona cancel, se cierra la ventana de aviso y se continúa en el listado.

Nombre	Descripcion	Alcance	Origen	Coste	Responsable	Fecha Alta	Fecha Baja		
Gestor Contenidos UPV	Desarrollo de un sistema de gestión de contenidos para la Universidad Politécnica de Valencia	Nivel 3	Optimizar el actual portal de la UPV	55000	Juan Garcia Gonzalez	01/03/2013		Editar	Eliminar
Sistema de Gestión de I+D+i.	Desarrollo de un sistema de gestión de I+D+i basado en la norma UNE 166.002.	Nivel 3	Ayuda para planificar, organizar y controlar las unidades de I+D+i	50000	Gema Martinez Gimeno	06/03/2012		Editar	Eliminar

## Gestión de Módulos:

En la aplicación, los proyectos se componen de módulos de manera que a la hora de gestionar un módulo, este irá siempre relacionado con un determinado proyecto, por tanto, un módulo nunca existirá por si solo.

La opción de alta de un nuevo módulo lleva a un formulario donde se podrán introducir los campos asociados a dicho módulo, en su caso, Nombre, Descripción, Proyecto al que pertenece, mediante una lista desplegable con todos los proyectos existentes en el sistema, Fecha de inicio, por defecto sería la fecha actual Fecha de finalización del proyecto y Observaciones.

El campo Nombre y Proyecto son obligatorios.



**Nuevo Modulo**

Nombre: \*

Descripción:

Proyecto: \*

Fecha inicio:  

Fecha fin:  

Observaciones:

La opción de consulta de nuevo módulo permite introducir diferentes criterios de búsqueda, mostrando por defecto un listado de todos los módulos existentes en el sistema.

Existe la posibilidad de mostrar en el listado los módulos dados de baja haciendo click en el check “Incluir módulos dados de baja”.

**Consulta Módulos**

Criterios de búsqueda

Nombre:

Descripción:

Proyecto: Selecciona una opción

Fecha Inicio:

Fecha Fin:

Incluir módulos dados de baja

**Listado Módulos**

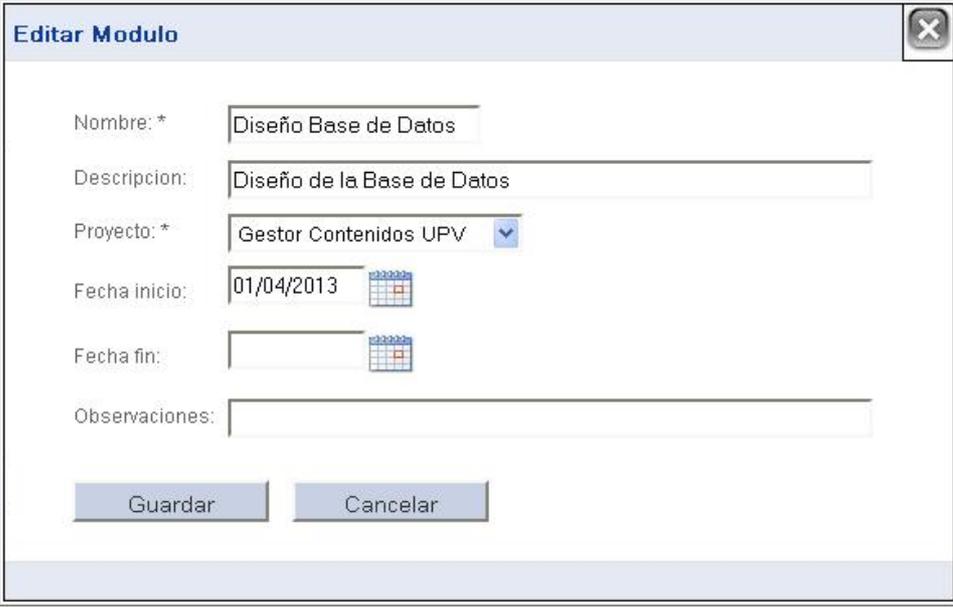
Nombre	Descripción	Proyecto	Fecha Alta	Fecha Baja	Observaciones
Análisis + Diseño	Describir los requisitos de forma entendible en un entorno web	Gestor Contenidos UPV	21/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Diseño Base de Datos	Diseño de la Base de Datos	Sistema de Gestión de I+D+i	01/04/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Especificación de Requisitos	Describir las necesidades del proyecto a partir de la propuesta y de las opiniones del usuario	Gestor Contenidos UPV	01/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementación	Elaboración del prototipo funcional y su posterior desarrollo	Gestor Contenidos UPV	02/04/2013	20/09/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>
Mantenimiento	Puesta en marcha del proyecto y atención a las necesidades de cambios	Gestor Contenidos UPV	09/05/2012		<a href="#">Editar</a> <a href="#">Eliminar</a>
Pruebas	Pruebas unitarias, de integración y de aceptación de usuario	Gestor Contenidos UPV	23/09/2013	12/12/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>

1-6 of 6

Para cada módulo del listado, se muestran dos links, Editar y Eliminar.

Si se hace click sobre Editar, el sistema abre una ventana de pop up con un formulario relleno con los datos del módulo seleccionado.

El usuario tiene la opción de modificar cualquier campo del módulo y pulsar sobre Guardar si desea guardar sus cambios o Cancelar para desecharlos. En ambos casos, la aplicación cierra el pop up y vuelve al listado de módulos anterior, actualizándolo.



The image shows a web application window titled "Editar Modulo" with a close button in the top right corner. The form contains the following fields:

- Nombre: \*** with the text "Diseño Base de Datos" entered in the input field.
- Descripcion:** with the text "Diseño de la Base de Datos" entered in the input field.
- Proyecto: \*** with a dropdown menu showing "Gestor Contenidos UPV".
- Fecha inicio:** with the date "01/04/2013" and a calendar icon.
- Fecha fin:** with an empty input field and a calendar icon.
- Observaciones:** with an empty text area.

At the bottom of the form, there are two buttons: "Guardar" and "Cancelar".

Si el usuario hace click sobre Eliminar, el sistema muestra un mensaje de confirmación de eliminación del módulo en primera plana, con las opciones ok y cancel. Si el usuario selecciona ok, el módulo es eliminado del sistema y se vuelve al listado de módulos, habiéndose suprimido de éste el módulo seleccionado. Si el usuario selecciona cancel, se cierra la ventana de aviso y se continúa en el listado.

Consulta Módulos

Criterios de búsqueda

Nombre:

Descripción:

Esta seguro que desea eliminar el modulo?

OK Cancel

Buscar Limpiar

Listado Módulos

Nombre	Descripción	Proyecto	Fecha Alta	Fecha Baja	Observaciones	Editar	Eliminar
Análisis + Diseño	Describir los requisitos de forma entendible en un entorno web	Gestor Contenidos UPV	21/03/2013			Editar	Eliminar
Diseño Base de Datos	Diseño de la Base de Datos	Sistema de Gestión de I+D+i	01/04/2013			Editar	Eliminar
Especificación de Requisitos	Describir las necesidades del proyecto a partir de la propuesta y de las opiniones del usuario	Gestor Contenidos UPV	01/03/2013			Editar	Eliminar
Implementación	Elaboración del prototipo funcional y su posterior desarrollo	Gestor Contenidos UPV	02/04/2013	20/09/2013		Editar	Eliminar
Mantenimiento	Puesta en marcha del proyecto y atención a las necesidades de cambios	Gestor Contenidos UPV	09/05/2012			Editar	Eliminar
Pruebas	Pruebas unitarias, de integración y de aceptación de usuario	Gestor Contenidos UPV	23/09/2013	12/12/2013		Editar	Eliminar

1-6 of 6

## Gestión de Tareas:

Al igual que los módulos no pueden existir por sí solos, las tareas no pueden existir sin estar asociadas a un módulo determinado, es por ello que a la hora de insertar una nueva tarea, el sistema solicite como obligatorios, además del nombre de la tarea, el proyecto al que se quiere asociar dicha tarea. Una vez seleccionado el proyecto el sistema desplegará los módulos asociados a dicho proyecto. Cuando se hayan seleccionado tanto proyecto como módulo, el sistema permitirá la creación de la nueva tarea.

Al seleccionar Alta Tarea en el menú, el sistema muestra un formulario con los campos Nombre, Descripción, Proyecto al que pertenece, mediante una lista desplegable con todos los proyectos existentes en el sistema, Módulo al que pertenece, con la lista rellena con los módulos pertenecientes al proyecto, en caso de tener uno seleccionado, Fecha de inicio, por defecto sería la fecha actual Fecha de finalización del proyecto y Observaciones.

**Nueva Tarea**

Nombre: \*

Descripcion:

Proyecto: \*  ▼

Modulo: \*  ▼

Fecha inicio: 20/05/2012  

Fecha fin:  

Observaciones:

La opción de consulta de nueva tarea permite introducir diferentes criterios de búsqueda, mostrando por defecto un listado de todas las tareas existentes en el sistema.

Existe la posibilidad de mostrar en el listado las tareas dadas de baja haciendo click en el check "Incluir tareas dadas de baja".

**Consulta Tareas**

**Criterios de búsqueda**

Nombre:

Descripción:

Proyecto:

Modulo:

Fecha Inicio:

Fecha Fin:

Incluir tareas dadas de baja

**Listado Tareas**

Nombre	Descripción	Proyecto	Modulo	Fecha Alta	Fecha Baja	Observaciones
Diseño pruebas	Diseño de los casos de test	Gestor Contenidos UPV	Analisis + Diseño	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Diseño UI	Diseño Grafico de Ventanas	Gestor Contenidos UPV	Analisis + Diseño	30/03/2013	09/04/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>
Estimacion horas	Estimacion horas del resto de tareas	Gestor Contenidos UPV	Analisis + Diseño	08/03/2011	24/05/2011	<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementacion DAOs	Implementacion Acceso a BBDD	Gestor Contenidos UPV	Especificacion de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementacion Servicios	Implementacion Servicios	Gestor Contenidos UPV	Implementacion	30/03/2013	09/04/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementacion UI	Implementacion Ventanas	Gestor Contenidos UPV	Especificacion de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Testing	Realizacion de los test	Gestor Contenidos UPV	Especificacion de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>

Para cada tarea del listado, se muestran dos links, Editar y Eliminar.

Si se hace click sobre Editar, el sistema abre una ventana de pop up con un formulario relleno con los datos de la tarea seleccionada.

El usuario tiene la opción de modificar cualquier campo de la tarea y pulsar sobre Guardar si desea guardar sus cambios o Cancelar para desecharlos. En ambos casos, la aplicación cierra el pop up y vuelve al listado de tareas anterior, actualizándolo.



Editar Tarea

Nombre: \*

Descripcion:

Proyecto: \*

Modulo: \*

Fecha inicio:  

Fecha fin:  

Observaciones:

Si el usuario hace click sobre Eliminar, el sistema muestra un mensaje de confirmación de eliminación de la tarea en primera plana, con las opciones ok y cancel. Si el usuario selecciona ok, la tarea es eliminada del sistema y se vuelve al listado de tareas, habiéndose suprimido de éste la tarea seleccionada. Si el usuario selecciona cancel, se cierra la ventana de aviso y se continúa en el listado.

**Consulta Tareas**

Criterios de búsqueda

Nombre:

Descripción:

Incluir tareas dadas de baja

Esta seguro que desea eliminar la tarea?

**Listado Tareas**

Nombre	Descripción	Proyecto	Modulo	Fecha Alta	Fecha Baja	Observaciones
Diseño pruebas	Diseño de los casos de test	Gestor Contenidos UPV	Análisis + Diseño	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Diseño UI	Diseño Grafico de Ventanas	Gestor Contenidos UPV	Análisis + Diseño	30/03/2013	09/04/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementación DAOs	Implementación Acceso a BBDD	Gestor Contenidos UPV	Especificación de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementación Servicios	Implementación Servicios	Gestor Contenidos UPV	Implementación	30/03/2013	09/04/2013	<a href="#">Editar</a> <a href="#">Eliminar</a>
Implementación UI	Implementación Ventanas	Gestor Contenidos UPV	Especificación de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>
Testing	Realización de los test	Gestor Contenidos UPV	Especificación de Requisitos	30/03/2013		<a href="#">Editar</a> <a href="#">Eliminar</a>

1-6 of 6

## Gestión de Técnicos:

La opción de alta de un nuevo técnico lleva a un formulario donde se podrán introducir los campos asociados a dicho técnico, en su caso, Nombre, Apellidos, NIF, Perfil del técnico obtenido en un desplegable con todos los perfiles que existen en la organización, Fecha de inicio, por defecto sería la fecha actual Fecha de baja del técnico en la organización.

Los campos Nombre, Apellidos, NIF y Perfil son obligatorios.

El listado de perfiles se recupera de una tabla estática donde están almacenados todos los perfiles existentes en la organización, para añadir o eliminar perfiles, debería hacerse mediante la manipulación explícita de la tabla en la base de datos.

**Nuevo Técnico**

Nombre: \*

Apellidos: \*

NIF: \*

Perfil: \* Selecciona una opción

Fecha inicio:

Fecha Fin:

La opción de consulta de nuevo técnico permite introducir diferentes criterios de búsqueda, mostrando por defecto un listado de todos los técnicos existentes en el sistema.

Existe la posibilidad de mostrar en el listado los técnicos dados de baja haciendo click en el check "Incluir técnicos dados de baja".

**Consulta Técnicos**

Criterios de búsqueda

Nombre:

Apellidos:

NIF:

Incluir técnicos dados de baja

**Listado Técnicos**

Nombre	Apellidos	NIF	Puesto	Fecha Alta	Fecha Baja		
Juan	García Gonzalez	36444689K	Analista Funcional	10/10/2011		Editar	Eliminar
Miguel	Gil Gil	69875423F	Programador Senior	25/07/2011		Editar	Eliminar
Gema	Martinez Gimeno	68972135E	Jefe Proyecto	23/01/2012		Editar	Eliminar
Jose	Martinez Herrero	45875216R	Diseñador Grafico	09/05/2012		Editar	Eliminar
Maria	Perez Perez	2B	Programador Junior	01/10/2006		Editar	Eliminar
Francisco	Ruiz Zanon	66354778U	Programador Junior	07/05/2012		Editar	Eliminar

1-6 of 6

Para cada técnico del listado, se muestran dos links, Editar y Eliminar.

Si se hace click sobre Editar, el sistema abre una ventana de pop up con un formulario relleno con los datos del técnico seleccionado.

El usuario tiene la opción de modificar cualquier campo del técnico salvo su NIF y pulsar sobre Guardar si desea guardar sus cambios o Cancelar para desecharlos. En ambos casos, la aplicación cierra el pop up y vuelve al listado de técnicos anterior, actualizándolo.



The image shows a screenshot of a web application window titled "Editar Técnico". The window contains a form with the following fields and controls:

- Nombre: \***: Text input field containing "Miguel".
- Apellidos: \***: Text input field containing "Gil Gil".
- NIF: \***: Text input field containing "69875423F".
- Perfil: \***: Dropdown menu with "Analista Funcional" selected.
- Fecha inicio:**: Date input field containing "25/07/2011" with a calendar icon.
- Fecha fin:**: Date input field with a calendar icon.
- Guardar**: Button to save changes.
- Cancelar**: Button to cancel changes.

Si el usuario hace click sobre Eliminar, el sistema muestra un mensaje de confirmación de eliminación del técnico en primera plana, con las opciones ok y cancel. Si el usuario selecciona ok, el técnico es eliminado del sistema y se vuelve al listado de técnicos, habiéndose suprimido de éste el técnico seleccionado. Si el usuario selecciona cancel, se cierra la ventana de aviso y se continúa en el listado.

Consulta Tecnicos

Criterios de busqueda

Nombre:

Apellidos:

NIF:

Incluir tecnicos dados

Buscar Limpia

Esta seguro que desea eliminar el tecnico?

OK Cancel

Listado Tecnicos

Nombre	Apellidos	NIF	Puesto	Fecha Alta	Fecha Baja		
Juan	Garcia Gonzalez	36444689K	Analista Funcional	10/10/2011		Editar	Eliminar
Miguel	Gil Gil	69875423F	Programador Senior	25/07/2011		Editar	Eliminar
Gema	Martinez Gimeno	88972135E	Jefe Proyecto	23/01/2012		Editar	Eliminar
Jose	Martinez Herrero	45875216R	Diseñador Grafico	09/05/2012		Editar	Eliminar
Maria	Perez Perez	2B	Programador Junior	01/10/2006		Editar	Eliminar
Francisco	Ruiz Zanon	66354778U	Programador Junior	07/05/2012		Editar	Eliminar

1-6 of 6

## Gestión de Asignaciones:

Bajo la opción de menú de Tarea se encuentra el submenú Asignar Tarea.

Si se selecciona esta opción, el sistema muestra un formulario con las listas desplegables de Proyecto, Módulo, Tarea y Técnico y los campos Fecha de inicio de la asignación y Fecha de finalización, además de una serie de checks que permiten incluir los elementos dados de baja de las listas anteriores.

La lista desplegable de Proyecto muestra todos los proyectos existentes en el sistema. Si se selecciona un proyecto, el sistema muestra todos los módulos asignados a ese proyecto. De lo contrario, en la lista aparecen todos los módulos existentes en el sistema.

Del mismo modo, en la lista de Tareas, aparecen las tareas asociadas al módulo seleccionado en caso de haber uno, o todas las tareas existentes en el sistema en caso contrario.

En la lista desplegable de técnicos aparecen todos los técnicos dados de alta en el sistema.

Una vez seleccionados al menos la tarea y el técnico, el sistema permite la asignación, que pasará a mostrarse en el listado de la parte inferior de la pantalla.

**Asignar Tarea**

Proyecto:

Módulo:

Tarea: \*

Técnico: \*

Fecha inicio:

Fecha fin:

Incluir proyectos dados de baja

Incluir módulos dados de baja

Incluir tareas dadas de baja

Incluir técnicos dados de baja

**Listado Asignaciones**

Proyecto	Módulo	Tarea	Técnico	Fecha Alta	Fecha Baja		
Gestor Contenidos UPV	Análisis + Diseño	Diseño pruebas	Jose Martinez Herrero	09/05/2012		<a href="#">Editar</a>	<a href="#">Eliminar</a>
Gestor Contenidos UPV	Especificación de Requisitos	Implementación DAOs	Gema Martinez Gimeno	09/05/2012	22/05/2012	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Gestor Contenidos UPV	Especificación de Requisitos	Implementación DAOs	Juan Garcia Gonzalez	09/05/2012	18/05/2012	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Gestor Contenidos UPV	Especificación de Requisitos	Implementación DAOs	Miguel Gil Gil	10/05/2012		<a href="#">Editar</a>	<a href="#">Eliminar</a>
Gestor Contenidos UPV	Implementación	Implementación Servicios	Juan Garcia Gonzalez	09/05/2012		<a href="#">Editar</a>	<a href="#">Eliminar</a>

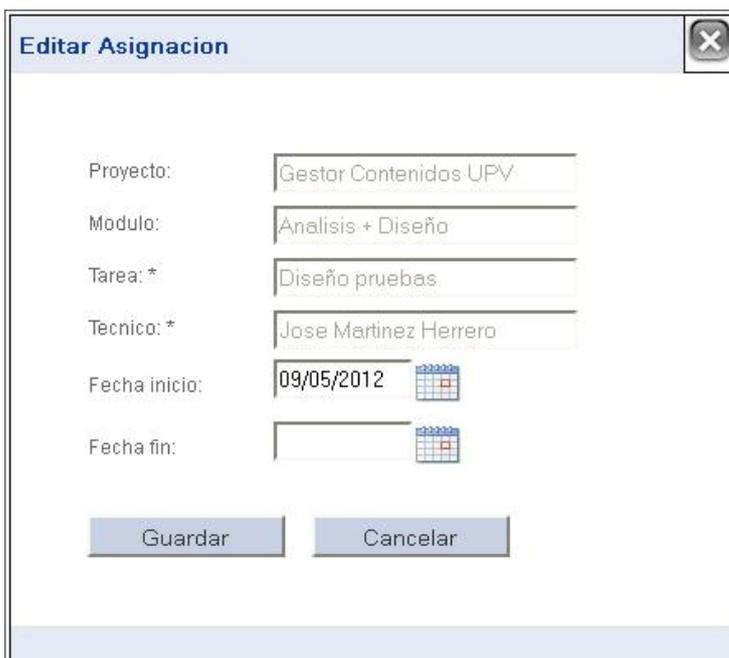
1-5 of 5

Para cada asignación del listado, se muestran dos links, Editar y Eliminar.

Si se hace click sobre Editar, el sistema abre una ventana de pop up con un formulario relleno con los datos de la asignación seleccionada.

El sistema sólo permite modificar la fecha de inicio y la de finalización de la asignación. En caso de querer cambiar el técnico o la tarea, deberá procederse a la eliminación de la asignación seleccionada y a la posterior creación de la nueva.

El usuario debe pulsar sobre Guardar si desea guardar sus cambios o Cancelar para desecharlos. En ambos casos, la aplicación cierra el pop up y vuelve al listado de asignaciones anterior, actualizándolo.

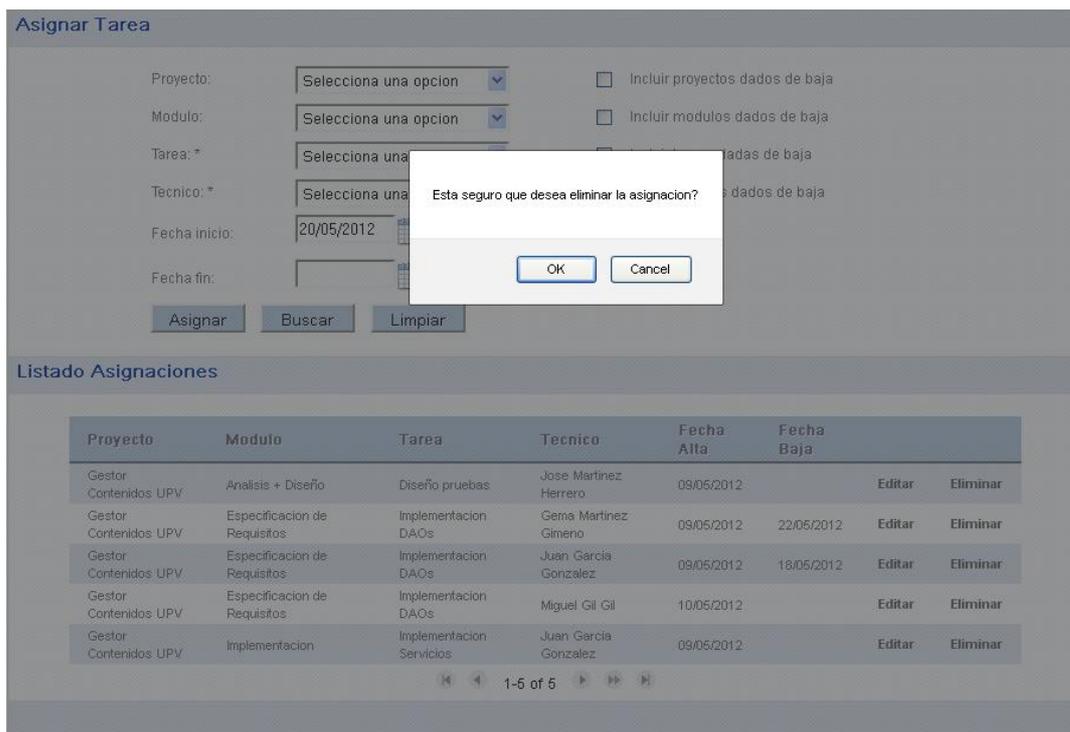


The image shows a dialog box titled "Editar Asignacion" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Proyecto:
- Modulo:
- Tarea: \*
- Tecnico: \*
- Fecha inicio:  
- Fecha fin:  

At the bottom of the dialog are two buttons: "Guardar" and "Cancelar".

Si el usuario hace click sobre el link Eliminar, el sistema muestra un mensaje de confirmación de eliminación de la asignación en primera plana, con las opciones ok y cancel. Si el usuario selecciona ok, la asignación es eliminada del sistema y se vuelve al listado de asignaciones, habiéndose suprimido de éste la asignación seleccionada. Si el usuario selecciona cancel, se cierra la ventana de aviso y se continúa en el listado.



Para todos los formularios de la aplicación, se tiene la opción de Limpiar que dejaría en blanco todos los campos del formulario en cuestión.

En todos los listados de la aplicación, existe paginación cuando el número de elementos en el listado es superior a 10.

Los iconos de calendario mostrados en la aplicación abren una ventana pop up con un calendario que permite seleccionar el día deseado al usuario.



### 3.4. PRUEBAS

El proceso de pruebas consiste, básicamente, en la realización de una serie de pruebas al código para observar con que frecuencia podría fallar ese código, encontrar esos errores y solucionarlos.

Estas pruebas hay que realizarlas al comienzo del ciclo de vida del software ya que, cuando se avanza en el ciclo de vida, más costoso sería solucionar el fallo encontrado.

Funcionalmente, por pruebas se entiende el análisis experimental que permite aumentar la confianza en que la implementación que una aplicación proporciona es correcta respecto a sus requisitos.

Las pruebas se pueden dividir en dos grupos:

1. Pruebas Unitarias, probamos el funcionamiento de cada módulo por separado.
2. Pruebas de Integración, probamos el funcionamiento del sistema.

#### Pruebas unitarias

Cuando cierta parte de un proyecto software está terminada, se debe realizar una serie de pruebas en busca de fallos a través de unos criterios llamados pruebas de *caja negra* y/o de *caja blanca*, no son excluyentes, sino complementarias, y ambos deben ser utilizados.

Si nos lo que importa son las devoluciones de cada una de las funciones de la CUT (Class Under Test), estaremos ante lo que se llaman *pruebas funcionales, de comportamiento* o de *caja negra*. Estas pruebas no están basadas en el conocimiento del código o

diseño interno y determinan la funcionalidad del sistema de acuerdo con los requisitos funcionales.

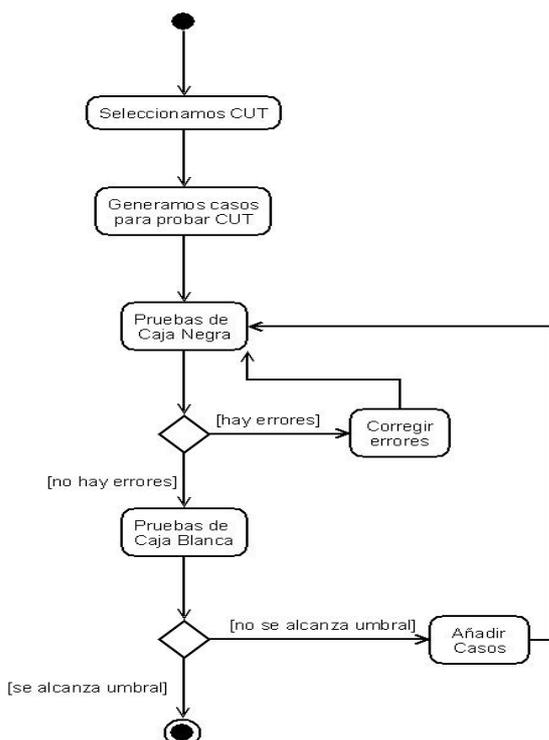
El proceso de una prueba de caja negra es simple, se ejecuta la unidad de prueba con datos, se observa la salida y se compara con el resultado esperado. Si la salida y el resultado esperado son iguales, la prueba ha sido superada.

Conforme se van pasando las pruebas de caja negra se puede determinar que “cantidad” de código ha sido cubierto, es decir, cuanto porcentaje de código se ha ejecutado. Esto son las pruebas de caja blanca o de implementación. Estas pruebas están basadas en la lógica interna de la aplicación y el código y se encargan de realizar una cobertura de declaraciones de código, ramas, caminos y condiciones.

Con las pruebas de caja blanca se busca encontrar fragmentos del programa que no son ejecutados por los casos de pruebas.

Si encontramos que el resultado de estas pruebas es menor al 100%, se debe ejecutar otros casos para intentar llegar al 100%. Si aun así no se consigue ese 100%, deberíamos preguntarnos si sirve de algo ese trozo de código.

El proceso de realizar estas pruebas es simple, como muestra el diagrama:



## Pruebas de integración

Se basan en las pruebas de conexiones y comunicaciones entre distintos módulos, una vez probados unitariamente. Este tipo de pruebas son esenciales en sistemas cliente-servidor o red. En este tipo de pruebas se combinan los módulos individuales y se prueban en grupo para asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

En concreto, para el proyecto realizado, se han ejecutado pruebas mayoritariamente dinámicas, esto es, ejecutando el código fuente para comprobar si las operaciones llevadas a cabo han sido reflejadas en la base de datos.

Al disponerse del código fuente de la aplicación, las pruebas unitarias han sido de caja blanca, intentando encontrar errores en los requisitos funcionales y de información y comprobando que el estilo de las interfaces se adapta a la resolución correspondiente.

También se han realizado pruebas de integración consistentes en probar la interacción de la aplicación con la base de datos.

Para facilitar la tarea de la ejecución de pruebas, se han utilizado clases con casos de prueba mediante JUnit.

JUnit es un conjunto de clases (*framework*) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Gracias a él, es posible automatizar las pruebas de las aplicaciones Java.

La única precaución que se debe de tener para utilizar JUnit es añadir la librería al classpath y tener el driver jdbc también añadido si se desea interactuar con la base de datos.

Algunas de las pruebas realizadas han sido:

- Comprobar que la base de datos devuelve los proyectos correctos al invocar la búsqueda.
- Comprobar que la aplicación no deja introducir una fecha de baja anterior a la fecha de alta de un técnico, mostrando un mensaje de aviso.
- Comprobar que se introducen los campos requeridos al dar de alta una nueva tarea.
- Comprobar que las tareas mostradas para un determinado módulo son las correctas.
- Comprobar que no es posible la eliminación de un módulo si tiene tareas asociadas, mostrando un mensaje de aviso.

## 4. CONCEPTOS TEÓRICOS

### 4.1. AJAX

#### 4.1.1. DEFINICIÓN AJAX

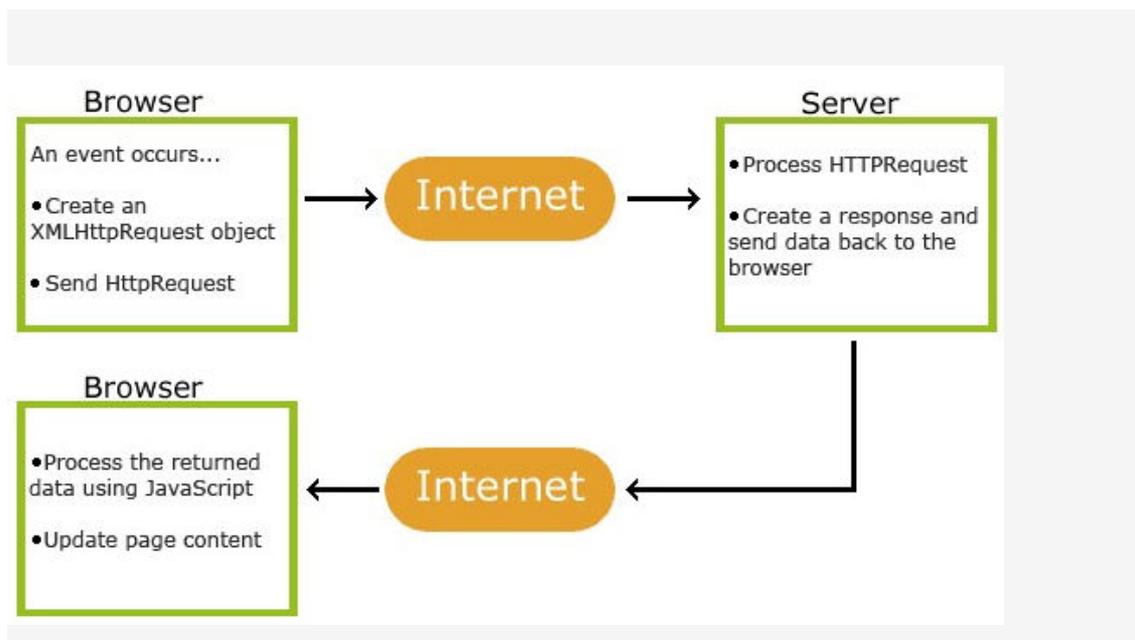
AJAX, es una técnica muy atractiva para muchos desarrolladores web de aplicaciones interactivas o RIA (Rich Internet Applications). Permite que los usuarios interactúen con un sitio y comunicarse con su base de datos sin actualizar las páginas. Sin embargo, debido a su naturaleza dinámica, las interfaces Ajax son a menudo más difíciles de desarrollar, en comparación con las páginas estáticas. A fin de reducir la complejidad del desarrollo, se han creado algunas bibliotecas de soporte de AJAX, como jQuery y GWT.

AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) es un grupo de técnicas de desarrollo web que se utilizan en el lado del cliente para crear aplicaciones web interactivas. Ajax permite a las aplicaciones web recuperar datos del servidor de forma asincrónica en segundo plano sin interferir con la visualización o el comportamiento de la página existente, lo que significa aumentar la interoperabilidad, velocidad y usabilidad en las aplicaciones.

AJAX no constituye una tecnología en si, sino que combina tres tecnologías ya existentes:

- XHTML y hojas de estilos (CSS) para el diseño que formatea la información.
- Document Object Model (DOM) que es el encargado de interactuar con la información presentada y es el que se ejecuta en el cliente (navegador), y
- XMLHttpRequest, que es un objeto encargado de intercambiar datos con el servidor web. Estos datos son devueltos en formato XML o JSON y se añaden a la página que estamos visualizando integrándose de nuevo gracias a XHTML y CSS.

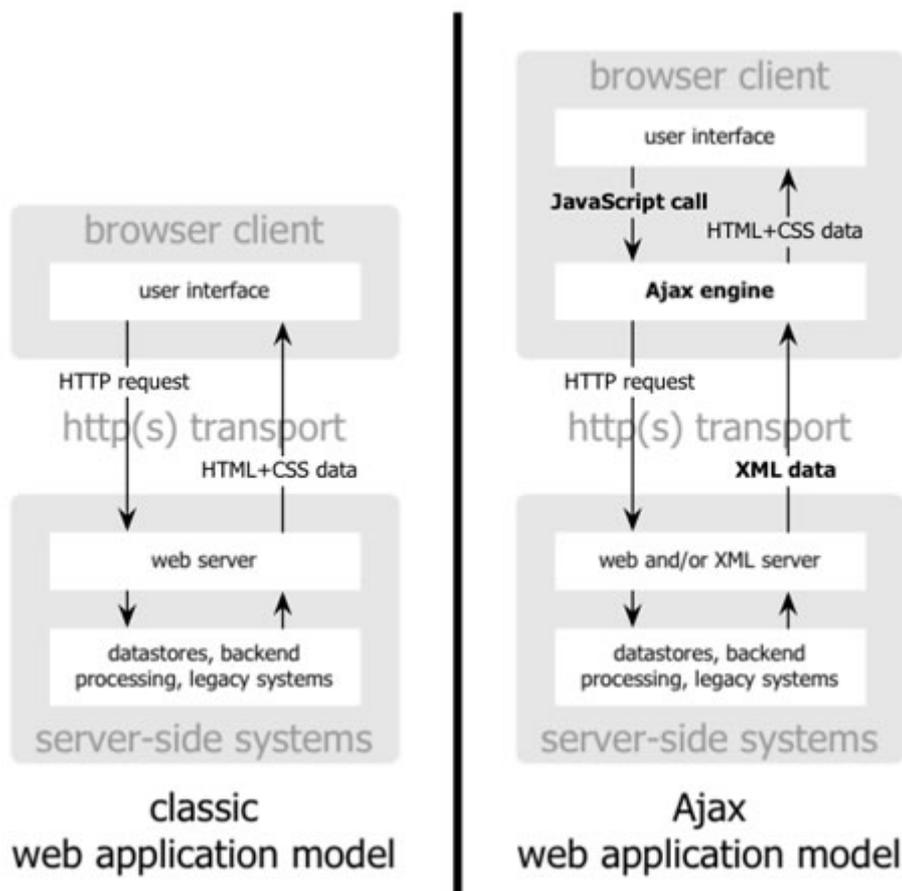
En el gráfico siguiente se muestra cómo trabaja Ajax:



El navegador está a la escucha de un evento y cuando éste se produce, crea un objeto XMLHttpRequest. El objeto XMLHttpRequest es el objeto que se utiliza para la comunicación entre servidor y el navegador en lugar de XMLHttpRequest, que actualizaría la página entera. La mayoría de los navegadores modernos tienen un XMLHttpRequest built-in, pero las versiones anteriores a IE6 no son compatibles con este objeto.

Si sólo hay una aplicación Ajax en la página, el concepto anterior se implementa fácilmente mediante un par de líneas en Javascript, pero cuando el proyecto incluye grandes usos de Ajax como Google Docs o Google Wave, el manejo de Javascript puede hacerse muy complejo.

Y en el siguiente, se muestra la diferencia entre las aplicaciones web clásicas y las aplicaciones web utilizando AJAX.



Las aplicaciones RIA son aplicaciones web con muchas de las características de las aplicaciones de escritorio, normalmente entregadas ya sea por medio de webs basadas en los estándares de los navegadores, vía plugins del navegador o independientemente, vía sandboxes o máquinas virtuales.

Desarrollar aplicaciones RIA utilizando Javascript tiene una serie de inconvenientes:

- Conseguir que el código sea cross-browser (funcione sin problemas en la mayoría de navegadores).
- Modularización del código cuando las aplicaciones crecen.
- Falta de herramientas avanzadas para el desarrollo con Javascript.
- Necesidad de tener un conocimiento avanzado en Javascript para obtener aplicaciones optimizadas.

Una solución a los problemas anteriores son los frameworks Javascript, como jQuery, Mootools, Prototype, etc. Pero aunque estos frameworks solucionan algunos de los problemas anteriores y pueden ser buena solución en muchos casos, no solucionan todos los problemas.

Para mejorar esto, en Google decidieron realizar una nueva caja de herramientas para acelerar el desarrollo de aplicaciones RIA basadas en AJAX mejorando la experiencia de usuario.

## **4.2. GWT**

### **4.2.1. DEFINICIÓN GWT**

GWT (Google Web Toolkit) es un conjunto de herramientas que permiten crear aplicaciones AJAX utilizando el lenguaje de programación Java que son compiladas posteriormente por el compilador de GWT en código Javascript optimizado para los principales navegadores.

Actualmente, la creación de aplicaciones web resulta un proceso pesado y propenso a errores. Los desarrolladores pueden pasar el 90% de su tiempo estudiando las peculiaridades de los navegadores. Por otra parte, la creación, la reutilización y el mantenimiento de una gran cantidad de componentes AJAX y bases de código JavaScript pueden ser tareas complejas y delicadas. Google Web Toolkit (GWT) facilita estas arduas tareas al ofrecer a los desarrolladores la posibilidad de crear y mantener rápidamente aplicaciones JavaScript con interfaces complejas, pero de gran rendimiento, en el lenguaje de programación Java.

El corazón de GWT es un compilador Java a JavaScript que produce código capaz de ejecutarse en Internet Explorer, Firefox, Safari y Opera. El compilador convierte la sintaxis Java a JavaScript,

utilizando versiones JavaScript de las clases Java más usadas como Vector, HashMap o Date.

Más allá del compilador, GWT incluye una extensa librería de widgets y paneles, haciendo más fácil desarrollar una aplicación web que luzca como una aplicación de escritorio. La librería incluye campos de texto, combos, barras de menú, árboles, diálogos, tabs, y otros.

Las aplicaciones generadas por GWT ejecutan código Java del lado del servidor, utilizando RPC para la comunicación entre el Cliente y el Servidor, llevando a cabo llamadas asíncronas.

#### **4.2.1.1. VENTAJAS**

##### **Crea aplicaciones AJAX en lenguaje Java y las compila a código Javascript optimizado para varios navegadores:**

A diferencia de los minimizadores de JavaScript, que solo funcionan con texto, el compilador de GWT realiza optimizaciones y un análisis estático completo de toda la base de código de GWT y, frecuentemente, genera código JavaScript que se carga y ejecuta con mayor rapidez que el código JavaScript equivalente creado de forma manual. Por ejemplo, el compilador de GWT suprime de forma segura todo el código no utilizable (mediante una exhaustiva tarea de eliminación de clases, métodos, campos, e incluso parámetros, que no se utilizan) para asegurarse de que el archivo de secuencias de comandos compilado sea lo más pequeño posible. Además, GWT compilará el código Java en archivos JavaScript independientes sin formato que estarán disponibles a través de cualquier servidor web. Por tanto, las aplicaciones GWT admiten automáticamente los navegadores IE, Firefox, Mozilla, Safari y Opera sin necesidad de detectar el navegador ni utilizar un formato especial en el código.

Solo se tendrá que escribir el código una vez y GWT lo convertirá al formato JavaScript más adecuado para el navegador de cada usuario.

### **Permite editar código Java y visualizar los cambios inmediatamente sin tener que volver a compilarlo:**

Durante el desarrollo de una aplicación, se pueden ver inmediatamente los cambios realizados en el código mediante el navegador de modo alojado (hosted mode) de GWT. No es necesario que se vuelva a compilar el código en JavaScript ni que sea implementado en un servidor. Solo se tiene que realizar los cambios deseados y hacer clic en "Actualizar" en el navegador de modo alojado.

### **Comunicación con el servidor mediante una sencilla llamada RPC:**

GWT admite un conjunto indefinido de protocolos de transferencia, como JSON y XML, pero el mecanismo de llamada a procedimiento remoto (RPC) de GWT permite el establecimiento de comunicaciones Java de una forma especialmente sencilla y eficaz. Al igual que ocurre con el mecanismo de invocación de métodos remotos (RMI) tradicional de Java, tan solo hay que crear una interfaz que especifique los métodos remotos que se quieran ejecutar. Al realizar una llamada a un método remoto desde el navegador, el mecanismo RPC de GWT serializará automáticamente los argumentos, ejecutará el método adecuado en el servidor y anulará la serialización del valor de retorno del código cliente. El mecanismo RPC de GWT también es bastante sofisticado: permite gestionar jerarquías de clase polimórfica y ciclos de gráficos de objetos, e incluso generar excepciones durante el proceso.

### **Fácil uso del historial y del botón de retroceso del navegador:**

Las aplicaciones AJAX no necesitan desgastar el botón de retroceso del navegador. GWT hace que resulte más sencillo utilizar un sitio web con solo añadir el estado al historial del botón de retroceso del navegador.

### **Permite el uso de herramientas de desarrollo productivas:**

GWT utiliza Java, por lo que se pueden emplear herramientas de desarrollo Java (Eclipse, IntelliJ, JProfiler y JUnit) al crear aplicaciones AJAX. De esa forma, se permite la comprobación de tipo estático en lenguaje Java permite que los desarrolladores detecten una clase de errores de JavaScript (errores ortográficos, tipos no coincidentes) en el momento de la creación del código, no durante la ejecución del programa, lo que aumenta la productividad y reduce los errores. Además, se pueden aprovechar abstracciones y patrones de diseño orientados a objetos (OO) basados en Java. Por otra parte, la integración directa de GWT con JUnit permite efectuar la comprobación por unidad tanto en un depurador como en un navegador, e incluso comprobar por unidad las llamadas a procedimientos remotos (RPC) asíncronas.

#### **4.2.1.2. INCONVENIENTES**

##### **No es un enfoque estándar de HTML y Javascript:**

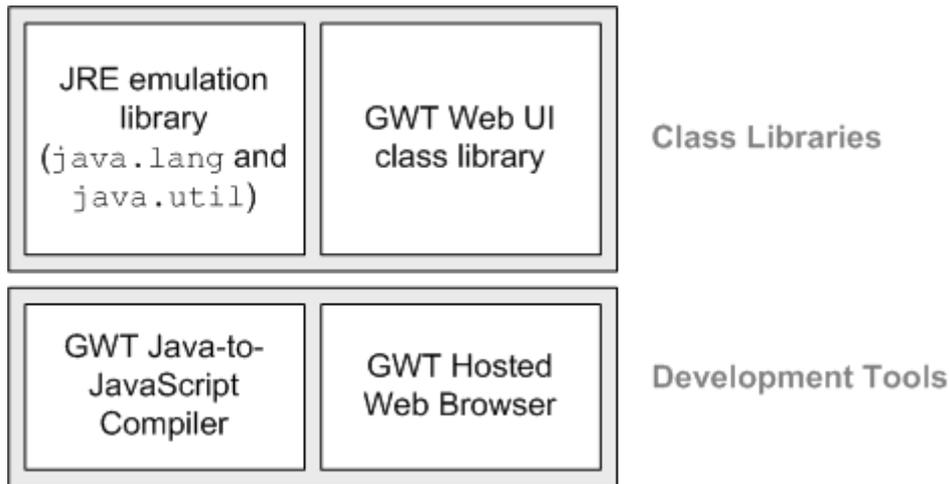
La generación automática de HTML podría hacer más difícil el uso de herramientas estándar de HTML y CSS.

##### **Sólo para desarrolladores Java:**

Muchos entornos AJAX permiten utilizar Javascript en el cliente y luego elegir el lenguaje en el servidor. GWT se encuentra basado enteramente en Java.

#### **4.2.2. ARQUITECTURA DE GWT**

GWT tiene cuatro componentes principales: un compilador Java-a-JavaScript, un navegador web "hosted", y dos librerías de clases:



- **Compilador GWT Java-a-JavaScript**

El Compilador GWT Java-a-JavaScript traduce del lenguaje de programación Java a JavaScript. El compilador se utiliza cuando se necesita ejecutar la aplicación en modo web.

- **Navegador web "Hosted" de GWT**

El Navegador web "Hosted" de GWT te permite ejecutar aplicaciones GWT en modo hosted, donde lo que se está ejecutando son bytecodes de Java sobre la máquina virtual de Java sin compilarlos a JavaScript. Utiliza Jetty como servidor embebido.

- **Emulación de librerías JRE**

GWT contiene implementaciones en JavaScript de las librerías de clases más usadas en Java, incluyendo la mayoría de las clases del paquete java.lang y un subconjunto de clases del paquete java.util. El resto del estándar de librerías de Java no es soportado nativamente con GWT. Por ejemplo, las clases de los paquetes como java.io no se utilizan en aplicaciones web ya que estas acceden a recursos en la red y al sistema de archivos local.

- **Librería de clases de interfaz de usuario de GWT**

Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que te permiten crear "widgets" para el navegador, como botones, cajas de texto, imágenes, y texto. Éste es el núcleo de las librerías de interfaz de usuario para crear aplicaciones GWT.

### 4.2.3. INTERFAZ DE USUARIO

El punto de inicio al crear una aplicación web utilizando GWT, es un archivo xml, en este caso, GestionProyectos.gwt.xml, que contiene la definición del módulo de GWT y sus dependencias. Es decir, las aplicaciones GWT se describen como módulos y los módulos se describen mediante archivos de configuración .gwt.xml. Cada módulo puede definir una o más clases de punto de entrada. Un punto de entrada es el punto de partida para una aplicación GWT, al igual que el método main de un programa estándar de Java. Por defecto, se utilizará una de las hojas de estilo heredadas de GWT a la que se hace referencia a través del tema heredado, en este caso, clean.css.

El aspecto que tendría este archivo sería el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>

<module rename-to='gestionproyectos'>
  <!-- Inherit the core Web Toolkit stuff. -->
  <inherits name='com.google.gwt.user.User' />
  <inherits name="com.google.gwt.i18n.I18N" />
  <inherits name="com.google.gwt.place.Place" />

  <!-- Inherit the default GWT style sheet. You can change -->
  <!-- the theme of your GWT application by uncommenting -->
  <!-- any one of the following lines. -->
  <inherits name='com.google.gwt.user.theme.clean.Clean' />

  <!-- Specify the app entry point class. -->
  <entry-point class='com.ejemplo.client.GestionProyectos' />

  <!-- Specify the paths for translatable code -->
  <source path='client' />
  <source path='shared' />

</module>
```

La página de inicio o página host, desde la que se inicia el código Javascript en nuestra aplicación, tiene el nombre de GestionProyectos.html. Aquí se hace referencia a la hoja de estilos propia de la aplicación, GestionProyectos.css.

La página de host hace referencia a la ruta de código fuente JavaScript (generados por GWT) responsable de los elementos dinámicos de la página.

El contenido completo del elemento body puede ser generado de forma dinámica. En este caso, se ha creado un elemento <div> HTML para usarlo como marcador de posición para los componentes GWT generados dinámicamente, que serán asignados al cuerpo de la etiqueta de la página HTML.

El aspecto de este archivo sería algo similar a:

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-
8">

    <link type="text/css" rel="stylesheet"
href="GestionProyectos.css">

    <title>Gestion Proyectos</title>

    <script type="text/javascript" language="javascript"
src="gestionproyectos/gestionproyectos.nocache.js"></script>
  </head>

  <!-- The body can have arbitrary html, or      -->
  <!-- you can leave the body empty if you want -->
  <!-- to create a completely dynamic UI.       -->

  <body>

  <h1>Sistema de Gestion de Proyectos</h1>

  <table id="main" style="height: 100%; width: 100%; "
cellpadding="0" cellspacing="0">

    <tr>
      <td id="menu" style="width: 100%; "></td>
    </tr>

    <tr>
      <td valign="top">
        <table width="100%" cellspacing="0" cellpadding="0"
border="0">
          <tr>
            <td id="gwt_content"></td>
          </tr>
        </table>
      </td>
    </tr>

    <tr>
      <td align="right" nowrap>
        <div class="footer"></div>
      </td>
    </tr>

  </table>
</body>
</html>
```

Una clase de Java que es punto de entrada debe implementar la interfaz "com.google.gwt.core.client.EntryPoint", que define el método de onModuleLoad (), en este caso, la clase que contiene el código fuente de Java para la aplicación de arranque es GestionProyectos.java.

Como la clase GestionProyectos.java ha sido especificada como la clase de punto de entrada en la definición del módulo GestionProyectos, cuando se lance GestionProyectos, se llamará al método onModuleLoad, que contiene la inicialización de la aplicación.

Para construir la interfaz de usuario, se ha utilizado UIBinder, que permite diseñar las interfaces de usuario de forma declarativa a través de XML.

Un ejemplo de código que usa UIBinder sería:

```
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.uibinder"
  xmlns:g="urn:import:com.google.gwt.user.client.ui">
  <ui:style>
    .mainPanel {
      border: 1px solid #e0e0e0;
    }
  </ui:style>
  <g:VerticalPanel height="100%" width="600px"
  horizontalAlignment="ALIGN_CENTER" styleName='{style.space_panel}'>

    <g:Grid width="250px" height="291px" ui:field="mainGrid" >
      <g:row>
        <g:customCell>
          <g:Label text="Nombre: *" width="85px" height="18px"
  styleName="text_label"/>
        </g:customCell>
        <g:customCell>
          <g:TextBox width="150px" height="18px" styleName="textbox"
  ui:field="nameField"/>
        </g:customCell>
      </g:row>
      . . .
    </g:Grid>

    <g:HorizontalPanel width="475px" height="54px"
  ui:field="createButton" >
      <g:Button width="80px" height="26px" text="Crear"
  styleName='{style.gpButton}' ui:field="saveButton" />
    </g:HorizontalPanel>

  </g:VerticalPanel>
</ui:UiBinder>
```

Las plantillas UiBinder tienen asociada una clase Java propietaria que permite el acceso a los widgets y paneles declarados en la plantilla, en este caso sería:

```
public class CrearProyectoImpl extends GPPanel implements
CrearProyecto {

    private static CrearProyectoImplUiBinder uiBinder =
GWT.create(CrearProyectoImplUiBinder.class);

    @UiField TextBox nameField;
    @UiField HorizontalPanel createButton;

    interface CrearProyectoImplUiBinder extends UiBinder<Widget,
CrearProyectoImpl> {}

    public void initUI() {
        setWidget(uiBinder.createAndBindUi(this));
        . . .
    }

    @UiHandler("saveButton")
    void onSaveButtonClick(ClickEvent event) {
        . . .
    }
}
```

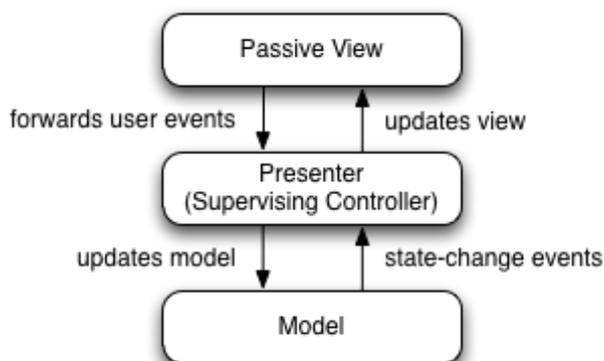
Cualquier objeto declarado en el archivo de ui.xml, incluidos todos los elementos DOM, puede ponerse a disposición de la clase Java propietaria a través del nombre del campo. Los elementos que se quieren tener accesibles desde el código Java, se marcan como ui:field en el ui.xml. Cuando uiBinder.createAndBindUi (this) se ejecuta, el campo anotado como @UiField en la clase Java se llena con la instancia adecuada del elemento.

La anotación @ UiHandler agrega un controlador al botón de GWT y sólo se tiene que escribir lo que se deberá ejecutar después que el botón mande un evento click.

## 4.2.4. EL PATRÓN MODELO-VISTA-PRESENTADOR

### 4.2.4.1. DEFINICIÓN

Este proyecto se basa en el patrón MVP, Model-View-Presenter, que es un patrón derivado del patrón Modelo Vista Controlador (MVC) que ayuda a ofrecer una clara separación entre la vista, el modelo y el controlador y cuyo esquema sería el siguiente:



La clave del patrón MVP es una estricta regulación de la interacción entre la vista y el controlador, aunque en el patrón MVP, al controlador se le conoce como presentador. La idea básica es que la clase Presenter haga de intermediario entre la Vista (la interfaz gráfica de usuario) y el modelo de datos y es ahí donde se realiza toda la lógica, que no depende en absoluto de los componentes de la interfaz gráfica y que, por tanto, es más fácil de realizar pruebas.

#### **4.2.4.2, IMPLEMENTACIÓN DEL PATRÓN MVP: ACTIVITIES Y PLACES**

En este proyecto, la implementación de este patrón se consigue mediante la utilización de Activities y Places, introducido a partir de la versión GWT 2.1, que además proporciona un marco integrado para la gestión del historial del navegador, permitiendo que el botón atrás del navegador funcione como el usuario espera, aunque en este proyecto, el mecanismo de control del historial del navegador no ha sido utilizado.

Una "Activity" (presentador) representa simplemente algo que el usuario está haciendo. Una actividad no contiene widgets o código de interfaz de usuario.

Las actividades suelen restaurar el estado ("despertar"), realizar la inicialización ("set up"), y cargar una interfaz de usuario correspondiente ("aparecer"). Las actividades se inician y se detiene por un `FragmentManager` asociado con un widget contenedor.

Una actividad puede mostrar automáticamente una confirmación de advertencia cuando la ésta está a punto de ser detenida (por ejemplo, cuando el usuario navega a un nuevo lugar). Además, el `FragmentManager` avisa al usuario antes de la ventana está a punto de ser cerrada.

Un ejemplo de esto lo encontramos en el siguiente código:

```
public class CrearProyectoActivity implements
CrearProyecto.Presenter{

    private CrearProyecto view = null;
    private CrearProyectoPlace place = null;

    public CrearProyectoActivity(final CrearProyectoPlace place) {
        this.place = place;
    }

    public void start() {
        view = new CrearProyectoImpl();
        view.setPresenter(this);
        Utils.changePage((CrearProyectoImpl) view);
        ((CrearProyectoImpl) view).initUI();
    }

    @Override
    public void onSaveButtonClick(ProyectoBean projectBean) {
        ProyectoServiceAsync rpc;
        rpc = (ProyectoServiceAsync) GWT.create(ProyectoService.class);

        rpc.createProject(projectBean, new AsyncCallback<Boolean>() {

            public void onSuccess(Boolean result) {
                if(result) {
                    Window.alert("Proyecto creado correctamente.");
                    ConsultarProyectoActivity activity = new
ConsultarProyectoActivity(new ConsultarProyectoPlace());
                    activity.start();
                }
            }

            public void onFailure(Throwable caught) {
                System.out.println("Error!! ");
                super.onFailure(caught);
            }
        });
    }
}
```

Lo primero a notar es que `CrearProyectoActivity` hace referencia a `CrearProyecto`, que es una interfaz de vista, no una implementación. Un

estilo de MVP codificación define la interfaz de vista en el presentador. Esto es perfectamente legítimo, sin embargo, no hay ninguna razón fundamental por la cual una actividad y una interfaz de vista correspondiente tengan que estar estrechamente unidos. Hay que tener en cuenta que `CrearProyectoActivity` también implementa la interfaz de la vista del `Presenter`. Esto se utiliza para permitir a la vista llamar a los métodos de la actividad, lo que facilita el uso de `UiBinder`.

El constructor `CrearProyectoActivity` toma como argumento `CrearProyectoPlace`.

El `CrearProyectoPlace` simplemente hace que sea fácil para `CrearProyectoActivity` para obtener las propiedades del estado representado por `CrearProyectoPlace`. Las actividades están diseñadas para ser desechables, mientras que las vistas, que son más caras de crear debido a las llamadas DOM necesarias, debe ser reutilizable.

Cuando se invoca al método `start`, las cosas se ponen en movimiento. En él se actualiza la vista y luego se redirige a ella.

Un "Place" es un objeto Java que representa un estado particular de la interfaz de usuario. Un place puede convertirse a y desde un token de historia URL mediante la definición de un `PlaceTokenizer` para cada place, y el `PlaceHistoryHandler` actualiza automáticamente la URL del navegador correspondiente a cada place en la aplicación.

Un place extiende de `com.google.gwt.place.shared.Place` y debe tener un `PlaceTokenizer` asociado que sabe cómo serializar el estado del place a un token URL. Por defecto, la dirección consiste en el nombre del place de clase simple (como "`CrearProyectoPlace` ") seguido de dos puntos (`:`) y el token devuelto por el `PlaceTokenizer`.

Un ejemplo de esto sería:

```
import com.google.gwt.place.shared.Place;

public class CrearProyectoPlace extends Place{

    public static class Tokenizer implements
PlaceTokenizer<CrearProyectoPlace> {
        @Override
        public CrearProyectoPlace getPlace(String token) {
            return new CrearProyectoPlace();
        }
    }
}
```

```
    }  
  
    @Override  
    public String getToken(final CrearProyectoPlace place) {  
        return null;  
    }  
}
```

Es conveniente (aunque no es obligatorio) declarar `PlaceTokenizer` como una clase estática dentro de la que corresponda. Sin embargo, no es necesario tener un `PlaceTokenizer` para cada `place`. Muchos `places` simplemente declaran un `PlaceTokenizer` que devuelve un token nulo porque no precisan guardar el estado de la URL, como es este caso.

Una “View” es simplemente la parte de la interfaz de usuario asociada a una actividad. En el desarrollo MVP, una vista se define por una interfaz, que permite múltiples implementaciones de la vista basadas en las características del cliente (como móvil vs escritorio) y también facilita las pruebas unitarias ligeras, evitando el consumo de tiempo que genera `GWTTestCase`. No hay ninguna interfaz o clase en GWT que la vista deba implementar o extender, sin embargo, GWT 2.1 presenta una interfaz `IsWidget` que es implementada por la mayoría de los `Widgets`, así como los `Composite`. Es útil para las vistas extender `IsWidget` si de hecho, proporcionan un `Widget`.

La vista contiene todos los componentes de interfaz de usuario que componen nuestra aplicación. Esto incluye todas las tablas, etiquetas, botones, cuadros de texto, etc. Las vistas son responsables de la distribución de los componentes de interfaz de usuario y no tienen noción del modelo. Es decir, un punto de vista no sabe que está mostrando un proyecto, simplemente sabe que tiene, por ejemplo, 3 etiquetas, cuadros de texto 3, y 2 botones que se organizan en forma vertical.

Ejemplos de código de la vista ya se proporcionaron al describir la interfaz de usuario con `UiBinder`.

El Modelo incluye objetos de negocio, en nuestro caso, por ejemplo, tendríamos un `ProyectoBean.java`, una representación del proyecto que contendría el identificador del proyecto, el nombre, la descripción, el

alcance, el origen, el coste, las fechas de inicio y finalización y el técnico responsable.

## **4.2.5. COMUNICACIÓN CON EL SERVIDOR**

Todas las aplicaciones GWT se ejecutan como código JavaScript en el navegador web del usuario final. Con frecuencia, sin embargo, se desea crear algo más que una aplicación standalone en el lado del cliente. La aplicación tendrá que comunicarse con un servidor web, enviar solicitudes y recibir las actualizaciones.

GWT ofrece un par de maneras diferentes para comunicarse con un servidor vía HTTP:

Se pueden utilizar llamadas a procedimiento remoto, esto es, el framework GWT RPC para hacer llamadas de forma transparente a los servlets de Java y dejar que GWT cuide de los detalles de bajo nivel como la serialización de objetos. Alternativamente, se pueden usar clases genéricas HTTP que ofrece GWT para construir la petición, como RequestBuilder, y clases JSON y XML de cliente para procesar la respuesta.

Tanto si se utiliza GWT RPC como datos JSON vía HTTP, todas las llamadas realizadas desde la página HTML al servidor son asíncronas. Esto significa que no se bloquean mientras esperan que retorne la llamada hecha al servidor. El código que va a continuación de la llamada se ejecuta inmediatamente. Cuando la llamada se completa, el método de devolución de llamada (callback) que se especificó cuando se hizo la llamada se ejecutará.

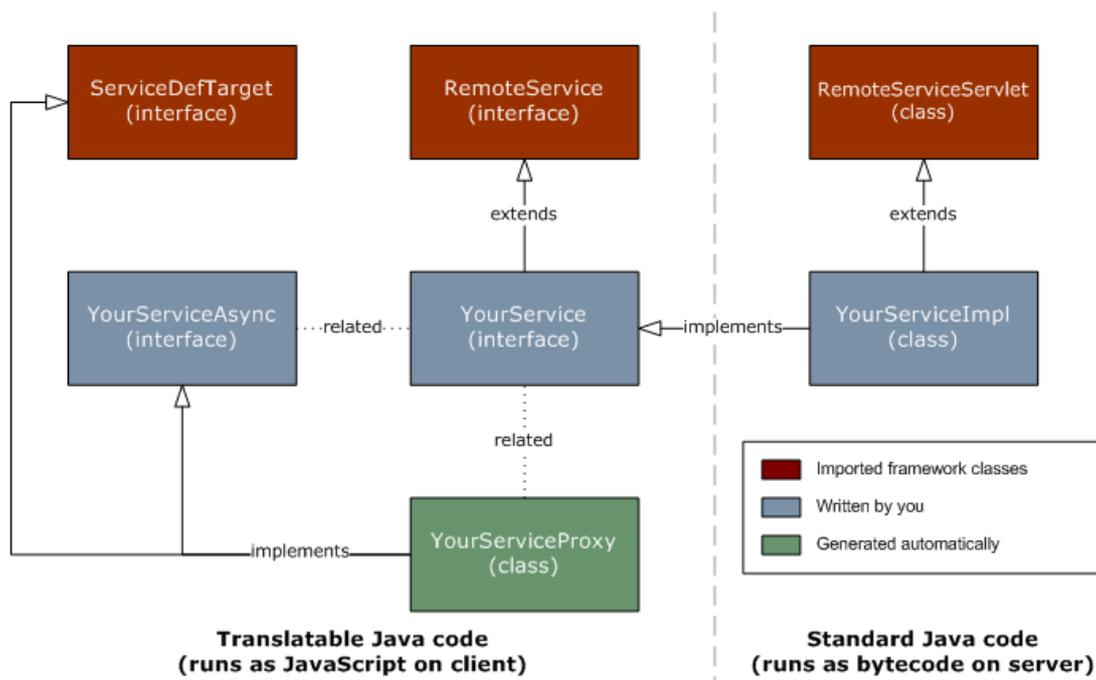
### **4.2.5.1. LLAMADAS RPC**

El mecanismo para interactuar con un servidor a través de una red se le llama hacer una llamada a procedimiento remoto (RPC), también referido a veces como llamada al servidor. GWT RPC hace que sea fácil para el cliente y el servidor pasar objetos Java de ida y vuelta a través de HTTP. Cuando se utiliza correctamente, RPC da la oportunidad de pasar toda la

lógica de la interfaz de usuario para el cliente, resultando en un rendimiento mejorado en gran medida, ancho de banda reducido, reducción de la carga del servidor web, y una experiencia de usuario agradable y fluida.

El código del lado del servidor que se invoca desde el cliente se refiere a menudo como un servicio, por lo que el acto de hacer una llamada a un procedimiento remoto se refiere a veces como la invocación de un servicio. Para ser claros, sin embargo, el servicio a largo plazo en este contexto no es el mismo que el concepto más general de "servicios web". En particular, los servicios de GWT no están relacionados con el Simple Object Access Protocol (SOAP).

Para invocar a un servicio, se necesitan una serie de elementos. Cada servicio tiene una pequeña familia de interfaces y clases de ayuda. Algunas de estas clases, como el proxy de servicio, se generan automáticamente y por lo general es como si no existieran. El modelo para las clases de ayuda es idéntico para todos los servicios que se implementan, y sigue el siguiente esquema:



Con el fin de definir una interfaz RPC, es necesario:

1. Definir una interfaz para el servicio que se extiende RemoteService y lista todos los métodos de RPC. Esta interfaz síncrona es la versión

definitiva de la especificación del servicio. Es el stub de cliente. No es posible llamar a esta versión del RPC desde el cliente, para ello se utiliza una interfaz asíncrona.

```
package com.ejemplo.client;

import com.google.gwt.user.client.rpc.RemoteService;
import com.google.gwt.user.client.rpc.RemoteServiceRelativePath;

@RemoteServiceRelativePath("proyecto")
public interface ProyectoService extends RemoteService {

    public boolean createProject(ProyectoBean project);

    . . .

}
```

2. Definir una clase para implementar el código del lado del servidor que extiende RemoteServiceServlet e implementa la interfaz que se ha creado anteriormente. Cada implementación del servicio es en última instancia, un servlet, pero en lugar de extender HttpServlet, extiende RemoteServiceServlet. RemoteServiceServlet maneja automáticamente la serialización de los datos que pasan entre el cliente y el servidor e invocar el método previsto en la implementación del servicio.

```
package com.ejemplo.server;

import com.ejemplo.client.ProyectoService;
import com.google.gwt.user.server.rpc.RemoteServiceServlet;

@SuppressWarnings("serial")
public class ProyectoServiceImpl extends RemoteServiceServlet
implements ProyectoService {

    public boolean createProject(ProyectoBean project){

        . . .

    }

}
```

3. Definir una interfaz asíncrona al servicio para llamar desde el código del lado del cliente. La naturaleza de las llamadas asíncronas requiere que el método que realiza la petición pase un objeto de devolución de llamada (callback) para que sea notificado cuando la llamada asíncrona se ha completado, ya que por definición quien llama no se puede bloquear hasta que la llamada se complete. Por la misma razón, los

métodos asincrónicos no tienen tipos de retorno, por lo general, devuelven void. Tras completarse la llamada asíncrona, toda la comunicación de vuelta al que realizó la llamada se realiza a través del objeto de devolución de llamada pasado.

```
package com.ejemplo.client;

import com.google.gwt.user.client.rpc.AsyncCallback;

public interface ProyectoServiceAsync {

    void createProject(ProyectoBean project, AsyncCallback<Boolean>
callback);

    . . .

}
}
```

Adicionalmente se debe modificar el archivo web.xml para anunciar el contexto por el cual las aplicaciones van a poder acceder al servicio (servlet) creado.

```
<servlet>
  <servlet-name>proyectoServlet</servlet-name>
  <servlet-class>com.ejemplo.server.ProyectoServiceImpl</servlet-
class>
</servlet>

<servlet-mapping>
  <servlet-name>proyectoServlet</servlet-name>
  <url-pattern>/gestionproyectos/proyecto</url-pattern>
</servlet-mapping>
```

La llamada al servicio desde el lado del cliente constaría de 3 pasos:

1. Instanciar la interfaz del servicio utilizando GWT.create().
2. Crear el objeto de retorno asíncrono (callback) que será notificado cuando la llamada RPC se haya completado.
3. Realizar la llamada.

```
public void onSaveButtonClick(ProyectoBean projectBean) {

    // 1. Crear el proxy cliente. El casting es necesario porque el
    // proxy generado implemente la interfaz asíncrona automáticamente
```

```
ProyectoServiceAsync rpc = (ProyectoServiceAsync)
GWT.create(ProyectoService.class);

// 2. Crear el objeto de retorno de llamada (callback) para manejar
el resultado
AsyncCallback callback = new AsyncCallback<Boolean>() {
    public void onSuccess(Boolean result) {
        // operaciones UI para manejar el éxito de la llamada
    }
    public void onFailure(Throwable caught) {
        // operaciones UI para manejar el error de la llamada
    }
};

// 3. Realizar la llamada.
// El flujo de control continua inmediatamente y más tarde se
invocará
// 'callback' cuando la llamada RPC se complete
rpc.createProject(projectBean, callback);
}
```

#### 4.2.5.2. MODOS DE EJECUCIÓN

En GWT existen dos modos de ejecución:

- Hosted mode: la aplicación se ejecuta como Java bytecodes en la Máquina Virtual de Java, y se emula un navegador. De este modo se puede trabajar en un entorno de desarrollo, como Eclipse, y aprovechar las ventajas de depurar código Java.
- Web mode: se ejecuta como puro JavaScript y HTML, compilado del Java original. Se ejecuta en un navegador externo, para comprobar cómo verá la aplicación el usuario final.

El modo hosted GWT con el servidor web embebido está diseñado únicamente para la depuración de la aplicación web. Una vez que la aplicación web está lista para su despliegue, es el momento de desplegarla en un servidor de producción. Si la aplicación web se compone sólo de contenido estático, casi cualquier servidor web servirá. Sin embargo, la mayoría de las aplicaciones web de GWT RPC usará servlets y Java. Para este tipo de aplicaciones, se tiene que

seleccionar un contenedor de servlets (también llamado un contenedor web o en el motor web) para ejecutar el back-end. GWT no proporciona un contenedor de servlets para utilizar en la producción, pero hay muchos productos diferentes disponibles como Apache Tomcat, Jetty, JBoss IBM WebSphere, etc.

## **5. CONCLUSIONES**

Una vez terminado el proyecto y cumplido todos los objetivos planteados, me siento capaz de realizar un portal Web con este framework, desde su configuración, su planificación, su desarrollo, su testeo y su documentación.

He adquirido más conocimientos acerca de un framework que cada vez está más extendido y es más demandado por la industria hoy en día, en cuanto a que ofrece la posibilidad de programar de forma más sencilla y facilitando enormemente la tarea de debug, al ser código java, aprovechando las ventajas que trabajar en este entorno ofrece. Además, al evitarse el uso de javascript escrito de forma manual, los típicos errores difíciles de encontrar y los problemas dependientes del navegador desaparecen. Esto me ha sido útil tanto personalmente como profesionalmente, ya que el día a día en mi trabajo me exige tener claros los conceptos de esta tecnología, sirviéndome el presente proyecto para asentarlos tanto de forma teórica como práctica.

He podido apreciar que utilizando GWT, en concreto la aproximación de Activities y Places aquí implementada ofrece la posibilidad de utilizar el patrón de diseño Modelo-Vista-Presentador, lo que implica una gran ventaja a la hora de codificar y separar en tareas la implementación de los proyectos que utilicen este patrón.

## **6. BIBLIOGRAFÍA**

Félix Buendía García, 2008: Una Guía para la realización y supervisión de Proyectos Final de Carrera (PFC) en el ámbito de la Web.

<http://dev.mysql.com>: Página oficial de MySQL.

<http://tomcat.apache.org>: Página oficial de Apache Tomcat.

<http://sourceforge.net/projects/eclipsesql>: Plugin SQL para eclipse.

<https://developers.google.com/web-toolkit>: Página oficial de GWT.

[www.epidataconsulting.com](http://www.epidataconsulting.com): Información sobre GWT.

[www.slideshare.net](http://www.slideshare.net): Información sobre GWT.

<http://hop2croft.wordpress.com>: Información sobre GWT.

<http://www.elrincondeajax.com>: Información sobre Ajax.

<http://www.codecompiling.net>: Información sobre Casos de Uso.

<http://www.adictosaltrabajo.com>: Información sobre pruebas y JUnit.

<http://www.forosdelweb.com>: Información variada sobre cómo conectar eclipse con MySQL.

<http://stackoverflow.com>: Información variada sobre cómo desplegar aplicaciones en Tomcat.