

UNIVERSIDAD POLITÉCNICA DE MADRID  
Escuela Universitaria  
de  
Ingeniería Técnica de Telecomunicación



PROYECTO FIN DE CARRERA

DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN  
PARA EL SEGUIMIENTO DE LA COOPERACIÓN  
UNIVERSIDAD-EMPRESA: SISCUE

MIGUEL ANGEL LUCAS DEL POZO

Abril de 2008



PROYECTO FIN DE CARRERA  
PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

TEMA: SISTEMAS DE INFORMACIÓN

TÍTULO: DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL SEGUIMIENTO DE LA COOPERACIÓN UNIVERSIDAD-EMPRESA: SISCUE

AUTOR: MIGUEL ANGEL LUCAS DEL POZO

TUTOR: ANA GOMEZ OLIVA

DEPARTAMENTO: DIATEL

Miembros del Tribunal Calificador:

PRESIDENTE: JUAN JIMENEZ TRILLO

VOCAL: ANA GÓMEZ OLIVA

VOCAL SECRETARIO: AURELIO BERGES GARCÍA

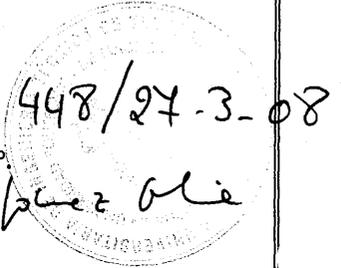
DIRECTOR:

Fecha de lectura: 7-4-08

Calificación: Sobresaliente 10p.

Vº Bº

Ana Gomez Oliva



El Secretario,

RESUMEN DEL PROYECTO:

SISCUE Es un sistema de información sobre la investigación de la UPM, realizado a partir de la necesidad de análisis sobre la cooperación universidad empresa. Consta de tres grandes bloques, una base de datos relacional (MySQL) con toda la información relativa a proyectos, investigadores, grupos de investigación y organismos, otro bloque de servicios en JAVA para explotar la base de datos, y por último, un interfaz web como acceso a dichos servicios por parte del usuario.

Entre los diferentes servicios que ofrece la aplicación explotados por el interfaz web, tenemos navegación y administración de los datos, exportación de datos a formatos típicos como PDF o EXCEL, y generación de gráficas con un servlet.

# DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL SEGUIMIENTO DE LA COOPERACIÓN UNIVERSIDAD-EMPRESA: SISCUE

<b>1</b>	<b>CAPÍTULO I el inicio: estudio para el seguimiento de cooperación universidad-empresa → proyecto acuse → siscue</b> .....	<b>4 -</b>
<b>2</b>	<b>CAPÍTULO II análisis y planteamiento de la situación inicial</b> .....	<b>7 -</b>
2.1	Objetivos buscados a partir del análisis (Acuse).....	7 -
2.2	Datos necesarios para el análisis y estado inicial de la información.....	8 -
2.3	Problemas básicos .....	10 -
2.4	Conclusiones: Descubriendo futuras posibilidades.....	10 -
<b>3</b>	<b>CAPÍTULO III búsqueda de una solución óptima</b> .....	<b>12 -</b>
3.1	Posibles maneras de abarcar los datos para un buen análisis.....	12 -
3.2	Abarcando el análisis desde un punto de vista técnico: Sistema de Información .....	15 -
<b>4</b>	<b>CAPÍTULO IV diseño y desarrollo del siscue</b> .....	<b>17 -</b>
4.1	Introducción, estructura del Siscue. ....	17 -
4.2	Bases de datos. ....	20 -
4.3	Plataforma .....	46 -
4.3.1	<i>Aplicación</i> .....	46 -
4.3.1.1	Diagrama general de la aplicación.....	49 -
4.3.1.2	Descripción de los módulos. ....	51 -
4.3.1.2.1	Módulo Acuse .....	52 -
4.3.1.2.2	Módulo Consultas .....	55 -
4.3.1.2.3	Módulo ConsultasV2.....	62 -
4.3.1.2.4	Módulo GruposC .....	78 -
4.3.1.2.5	Módulo Exportacion.....	79 -
4.3.1.2.6	Módulo PlanCalidad.....	80 -
4.3.1.2.7	Módulo Servlet: GeneracionGraficasSiscue .....	96 -
4.3.2	<i>Interfaz Web</i> .....	107 -
4.3.2.1	Diagrama de navegación y hojas creadas: Un viaje por la web. -	108 -
4.3.2.2	Sistema de permisos .....	148 -
4.4	Resumen de elementos usados para la creación y mantenimiento del sistema-	150 -
<b>5</b>	<b>CAPÍTULO V resultados logrados a partir de siscue, conclusiones</b> .-	<b>151 -</b>
5.1	Realización del análisis.....	151 -
5.2	Ventajas del uso de un Sistema de Información .....	151 -
5.3	Conclusiones, un amplio abanico de posibilidades.....	152 -
<b>6</b>	<b>ANEXO I</b> .....	<b>155 -</b>
<b>7</b>	<b>resumen y guía de usuario</b> .....	<b>155 -</b>
A1.1	Introducción .....	155 -

A1.2	Instalación del sistema .....	- 156 -
A1.2.1	Instalación del servidor MySQL5 y la base de datos .....	- 156 -
A1.2.2	Instalación del Apache Tomcat 5.0 y despliegación de la plataforma Siscue .....	- 159 -
A1.3	Navegación básica por los datos .....	- 161 -
A1.3.1	Barra Superior .....	- 161 -
A1.3.2	Barra Lateral .....	- 163 -
A1.3.3	Elementos: .....	- 164 -
A1.3.4	Estadísticas .....	- 167 -
A1.3.5	Busquedas avanzadas de elementos .....	- 170 -
A1.3.5.1	Busqueda avanzada de proyectos .....	- 170 -
A1.3.5.2	Búsqueda de grupos de investigación.....	- 174 -
A1.3.6	Plan de calidad.....	- 179 -
A1.3.7	Cuadro de mando Europeo FP6.....	- 181 -
A1.4	Administración de los datos .....	- 181 -
A1.4.1	Modificación y borrado .....	- 182 -
A1.4.2	Agregado de elementos .....	- 183 -
	<b>Bibliografía y referencias .....</b>	<b>- 187 -</b>



## CAPÍTULO I

### EL INICIO: ESTUDIO PARA EL SEGUIMIENTO DE COOPERACIÓN

#### UNIVERSIDAD-EMPRESA → PROYECTO ACUSE → SISCUE

---

Aprovecho para iniciar este Proyecto Fin de Carrera citando parte del estudio del “Análisis de la Cooperación de la Universidad Politécnica con el Sector Empresarial”, dirigido por Gonzalo León Serrano, Vicerrector de investigación de la UPM, ya que este estudio es la lanzadera de la que nació Siscue:

*“A iniciativa del Centro para el Desarrollo Tecnológico Industrial (CDTI), de la Fundación COTEC para la Innovación Tecnológica y del propio Consejo Social de la Universidad Politécnica de Madrid (UPM), se decidió llevar a cabo un profundo estudio de la relación universidad-empresa, y concretamente UPM-empresa, analizando múltiples perspectivas y diferentes tipos de instrumentos, en el periodo 2003-2006 (Proyecto ACUSE).*

*El objetivo perseguido por las tres instituciones es común: fortalecer la colaboración entre la universidad y el sector industrial, tanto público como privado, considerado como elemento favorecedor del bienestar social. Para ello ha sido necesario conocer en profundidad todos los escenarios posibles donde existe ese contacto, así como su evolución temporal; por razones operativas y de disponibilidad de los datos necesarios se ha propuesto realizar el análisis con los datos correspondientes al período 2003-2006.*

*La definición de la metodología de trabajo ha resultado una tarea realmente compleja, debido a los múltiples instrumentos de cooperación universidad-empresa existentes, a las diferentes fuentes de financiación, la variedad de agentes intervinientes, los tipos de estructuras de investigación en la UPM, etc.*

*Para el análisis de los datos disponibles en los años indicados, ha sido fundamental el diseño y construcción de la herramienta denominada SISCUE, "Sistema de Información para el Seguimiento de la Cooperación Universidad-Empresa". Actualmente en las bases de datos de SISCUE podemos encontrar información de más de 4.000 proyectos de investigación, desarrollo tecnológico e innovación (I+D+i), más de 4.500 investigadores y más de 3.100 organismos relacionados con la UPM, de los que alrededor del 50% se trata de empresas. Para la recolección de los datos se ha contado con las oficinas de gestión de recursos, tanto del Rectorado de la UPM, así como de las distintas fundaciones existentes adscritas a los distintos centros de nuestra universidad.*



FIGURA 1-1: ACCESO AL SISTEMA SISCUE

*El trabajo a realizar tiene un objetivo de permanencia y actualización continua. Por ello el SISCUE será en el futuro, una herramienta fundamental para la toma de decisiones por parte de los responsables de la UPM. Pero más allá del análisis cuantitativo que realiza SISCUE."*

Es por ello por lo que esta herramienta ha tenido que ser ampliamente diseñada para no solo abarcar todos los objetivos actuales, sino dejar abiertas nuevas líneas de trabajo apoyadas en el sistema. Como se podrá ver en el Capítulo IV, la herramienta se divide en 3 grandes bloques, la base de datos, la plataforma web, y la aplicación que hace de interfaz entre ambas. Todo esto completa un Sistema de información que ha sido desarrollado apoyándose en software libre y sobre todo, compatible. La página web como interfaz humana

permite compatibilidad a prácticamente todos los sistemas operativos actuales, ya que algo como el código html y javascript es comprendido por prácticamente todos los exploradores (iexplorer, mozilla, opera, etc.) de los diferentes sistemas operativos. A esto se le suma la ventaja de que el usuario no necesita software adicional para el uso del sistema, lo que incrementa la sencillez y la facilidad de uso, que es uno de los aspectos más importantes que se buscan actualmente en el desarrollo de software orientado al usuario medio.

Para el desarrollo de este sistema se han tenido que programar los tres bloques por separado, el orden lógico de desarrollo es:

1. **Diseño e implementación de la base de datos completa:** Crear una base de datos que nos permita tener toda la información que necesitamos, se desarrollo en MySQL teniendo en cuenta todos los aspectos actuales y futuros de la información.
2. **Diseño y desarrollo de todos los módulos de la aplicación:** ¿Qué servicios queremos tener de la base de datos? Se desarrolló en java numerosos módulos para los diferentes tipos de servicios que se le pueden pedir a la base de datos (pedir información, consultas complejas, etc.), así como servicios adicionales (creación de gráficas a partir de resultados obtenidos de la base de datos, exportación de datos, etc).
3. **Diseño de la web:** Una vez tenemos todos los servicios que necesitamos, los explotamos con una web que permita al usuario hacer uso de ellos de forma cómoda y transparente,

A día de escritura de este libro, SISCUE está siendo ampliado en varios aspectos, un ejemplo de ello es la creación de un observatorio de investigación, impulsado desde el Vicerrectorado de investigación de la UPM.

#### **2.1 *Objetivos buscados a partir del análisis (Acuse)***

Los objetivos básicos de la realización de dicho análisis son conseguir una visión global de la situación actual de la UPM con el sector empresarial. Con esta visión se pretende generar un estudio de varios factores:

- Líneas de cooperación de la UPM que necesitan un cambio de enfoque.
- Reingeniería en ciertos “modelos de negocio” a partir de los resultados que se van haciendo vigentes año a año.
- Conseguir “captar” la evolución de toda esta información para, en un futuro, poder hacer un seguimiento de la UPM en todos estos sectores.
- Otros factores

La UPM tiene un gran papel como universidad en colaboración con la empresa, que crece cada año, y empieza a ser vital la necesidad de controlar el estado actual, y su evolución en el tiempo, para facilitar una toma de decisiones en el momento adecuado. Este análisis pretende dar a conocer, no solo la situación actual de la UPM, sino la necesidad de poder manejar las direcciones que está tomando en cuanto a su papel con el sector empresarial. Es más, si el análisis es completo y abarca todos los puntos de una manera objetiva, servirá para dar un enfoque mucho más amplio del inicial, ya no solo en la colaboración con la empresa, sino con todos los tipos de organismos, incluso una visión interna en la universidad.

## **2.2 Datos necesarios para el análisis y estado inicial de la información**

Para la realización de un análisis completo y fiable, es fundamental reunir, analizar y verificar todos los datos relativos a los objetivos perseguidos. Para ello se reunieron todos los existentes relacionados con los siguientes bloques:

- **Proyectos:** Todos los bloques de proyectos en los que participa la UPM.
- **Investigadores:** De la UPM y externos.
- **Organismos:** Todos los tipos de organismos existentes, no solo el sector empresarial. Se pretende recoger todos los tipos de colaboración con la UPM.
- **Grupos de Investigación:** No se contó en un inicio, pero su adaptación futura permitió ver nuevos puntos de análisis.

### **Proyectos**

Sobre todo los siguientes marcos de colaboración: Nacionales, internacionales, regionales, artículo 83, europeo, convenios, cátedras, licencias, título propio ... Intentando reunir todos ellos en un "proyecto tipo" para un análisis genérico, permitiendo la posibilidad de comparaciones entre unos y otros.

*¿De donde se obtuvo toda la información de estos proyectos?*

Del Rectorado de la UPM, así como de las distintas fundaciones existentes adscritas a los distintos centros de nuestra universidad y dentro del rectorado, para cada tipo de proyecto de los organismos responsables de su gestión. Algunos ejemplos de los orígenes de datos son:

- Fichero Excel con los proyectos firmados al amparo del Artículo 83, desglosado por años.
- Fichero Access con los proyectos subvencionados del Plan Nacional.

- Fichero Access con los proyectos europeos, clasificados por marcos de colaboración (FP5,FP6,etc).
- Fichero Words con Licencias por años.
- Fichero Excel de listado de personal de la UPM.
- Fichero Acces con todos los grupos de investigación.

Y muchos otros.

### **Organismos**

A diferencia de lo que sucede en los Proyectos, no existen bases de datos de los organismos con los que trabaja la UPM como tal, sino que en cada marco de colaboración nos encontramos su consorcio, teniendo, por ejemplo, en las bases de Artículo 83, qué empresa ha subcontratado a la UPM, e incluso, algunas veces, no se tiene muy controlada esa información si no es relevante bajo el punto de vista de la gestión.

### **Investigadores**

Uno de los elementos más importantes para el análisis son los propios investigadores, donde conocer su participación en los proyectos es de máxima importancia para obtener resultados. Al buscar información de la actividad de dichos investigadores, como pueden ser proyectos, nos encontramos con que cada fuente trata su contenido de forma independiente, eliminando la compatibilidad con el resto de bases de datos. Al ser así, la labor de unificar toda la información por investigador requiere una labor artesanal y muy costosa.

### **Grupos de Investigación**

Muy parecido a lo anteriormente mencionado, se encontró una base de datos en Access con los grupos de investigación y parte de la información relacionada (personal, ayudas a grupos, etc.) sin normalizar, satisfaciendo en

poca medida las necesidades actuales, y quedándose corto para objetivos más ambiciosos.

### **2.3 Problemas básicos**

Al reunir todos estos datos nos encontramos con un enorme obstáculo. *¿A qué se debe éste obstáculo?* Es fácil comprender que a la hora de reunir un volumen de datos considerable de diferentes fuentes, surjan problemas como los siguientes:

- 1) Cada origen está basado en los requerimientos específicos de su poseedor.
- 2) Falta de normalización en prácticamente todas las fuentes de datos.
- 3) Imposibilidad de realizar consultas a todos estos datos utilizando las herramientas típicas de origen (Excel, Access, etc).
- 4) Limitado (casi nulo) poder de cálculo.
- 5) Cruzado de los datos totalmente inviable.
- 6) Y uno que nos afecta considerablemente: No se puede realizar una visión global a partir de estas fuentes, debido a los casos anteriormente expuestos, con lo que nos impide poder realizar un análisis global.

Es fácil llegar a la conclusión de que hay que buscar una solución en la cual todos estos problemas sean resueltos, reincidiendo en que esto no solo es para poder realizar este análisis puntual, sino para poder realizar una mejorada gestión de toda esta información para objetivos futuros.

### **2.4 Conclusiones: Descubriendo futuras posibilidades**

Viendo el estado actual de toda la información requerida, y asumiendo la importancia de estos datos no solo para el análisis, sino para la ayuda de la gestión y toma de decisiones futuras del Vicerrectorado de Investigación de la UPM, basándonos en los problemas anteriormente descritos, podemos resumir lo siguiente:

- Necesidad de mejora en la gestión de los datos relativos a la investigación.
- Necesidad de algún tipo de sistema que nos permita hacer análisis globales sobre toda la información ya gestionada.
- Con estos dos puntos, dejar la posibilidad de crecerlos para nuevos propósitos → Escalabilidad requerida.

Es importante destacar que el hecho de observar la situación actual para un objetivo específico, nos hace ver las amplias posibilidades que tendría aplicar algún tipo de proyecto orientado a la resolución de todos estos problemas de una manera centralizada, abierta y escalable. Si conseguimos unificar todos los datos de algún tipo de forma que tengamos la capacidad de preguntarle lo que necesitemos, podríamos no solo habilitar una gestión mucho más efectiva, sino aprovecharnos de esta gestión para realizar, por ejemplo, un observatorio de investigación donde la gente podría consultar sus necesidades de una manera efectiva, así como apoyándonos en esta información poder realizar todo tipo de consultas estadísticas que nos ayudarán a tomar decisiones con una base más férrea.

## CAPÍTULO III

### BÚSQUEDA DE UNA SOLUCIÓN ÓPTIMA

---

#### **3.1 Posibles maneras de abarcar los datos para un buen análisis.**

Visto el estado inicial de toda la fuente de datos, se pueden barajar varias formas de abarcar su manejo para completar el análisis, siendo algunas de dichas formas, las descritas a continuación:

**1. Reunir todos los datos relativos a cada fuente por separado, y luego analizar sobre ellos.**

*En este tipo de análisis, es una de las soluciones más adoptadas, debido a su simplicidad y la no necesidad de un perfil técnico para realizarlo.*

▪ **Ventajas:**

- i. Facilidad a la hora de recapitular la información.

▪ **Desventajas:**

- i. Análisis pobre e incompleto.
- ii. Cierra todo tipo de posibilidad futura. Para cualquier cambio o ampliación del análisis, habría que rehacer todo.
- iii. Reutilización del trabajo realizado nula.

**2. Crear una base informática tipo Access, creando todos los elementos necesarios para el análisis.**

*Esto fue lo que se adoptó como solución en un principio, partiendo, en parte, de esta forma hacia la final.*

▪ **Ventajas:**

- i. Análisis más completo.
- ii. Organización de los datos actuales a un paso "más normalizado".

▪ **Desventajas:**

- i. Dificultad de normalización y reunión de los datos.
- ii. Poco escalable, su uso futuro es posible, pero para objetivos muy básicos.

3. Elaborar un **Sistema de Información** con la tecnología actual, combinando las posibilidades de los tipos de bases de datos informáticas actuales con las herramientas programables de acceso/manipulación a ellas.

*De esta forma el trabajo se amplía considerablemente, pero los resultados obtenidos son muy favorables, ya no solo para el análisis, sino para el crecimiento de muchos factores relevantes en organismos como, en este caso, el Vicerrectorado de investigación de la UPM.*

▪ **Ventajas:**

- i. Análisis inicial completo, pudiendo ser ampliado en cualquier sentido.
- ii. Posibilidad de tener una visión global gracias a la unificación de la información de una manera lógica.
- iii. Normalización de la información a un nivel avanzado.
- iv. **Gestión futura de los datos** con un sistema de estas características, mejorando todo lo relativo a la administración de este tipo de información.
- v. Este tipo de sistema son escalables, permitiendo ampliar objetivos aprovechando el trabajo ya realizado.

- **Desventajas**

- i. Dificultad de reunificación y normalización de datos muy grande.
- ii. Diseño y desarrollo del sistema complejo.
- iii. Mantenimiento del Sistema para su utilización futura

### 3.2 Abarcando el análisis desde un punto de vista técnico: Sistema de Información

Partiendo de la segunda posibilidad vista en el punto 3.1, se pudo ver la deficiencia en el estado de la información que había para toda la gestión de proyectos, grupos de investigación, investigadores, etc. Así que se buscó ampliar esta opción a algo que permitiera, no solo el análisis que se quería llevar a cabo, sino mejorar la gestión actual y poder tener una visión global en la evolución en el tiempo de estos datos.

Es por ello que se decidió la elaboración de un Sistema de información complejo, para ello, el trabajo a realizar se puede dividir en varios aspectos, resumidos en la figuras 3-1 y 3-2.

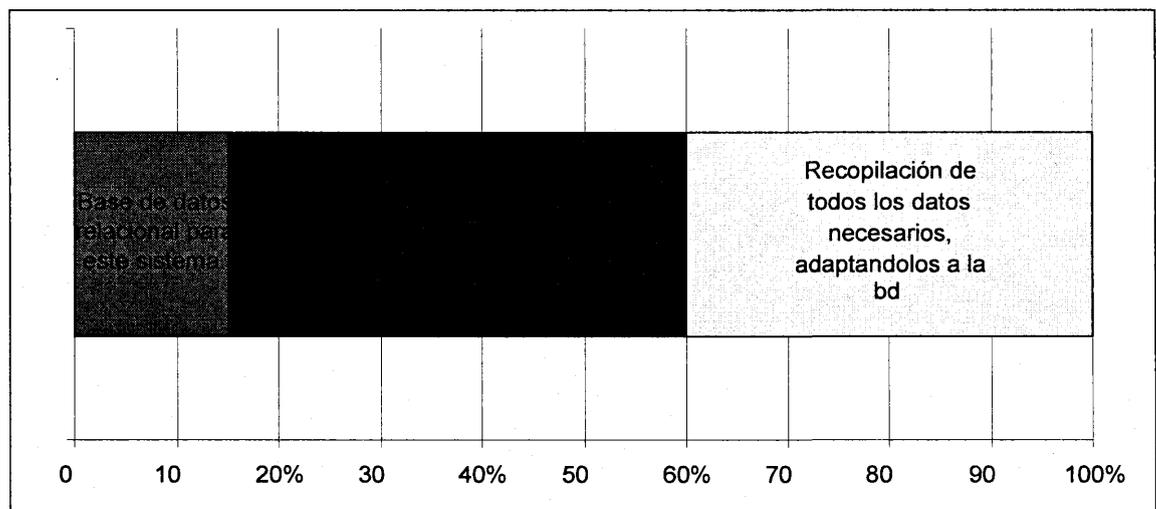


FIGURA 3-1: ESQUEMA IDEAL DE PROCESO DE TRABAJO

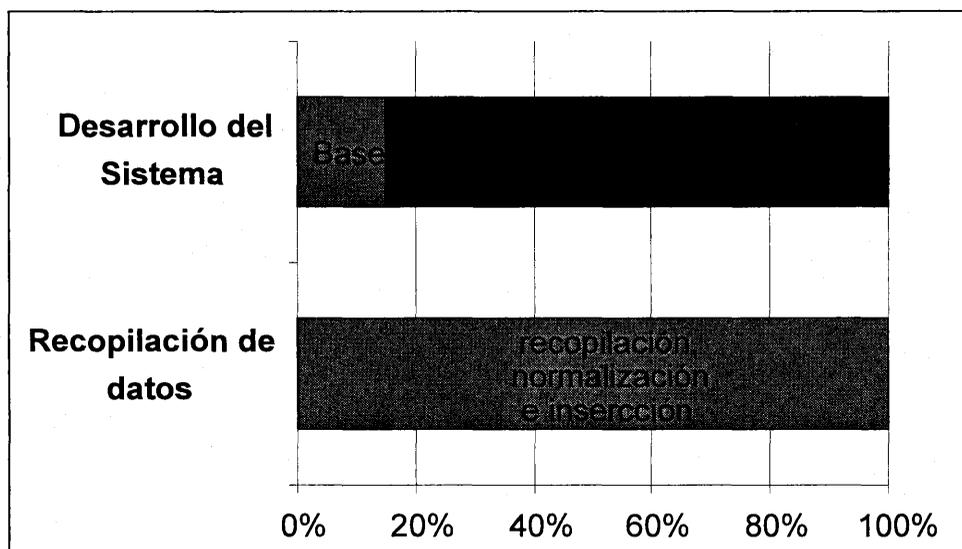


FIGURA 3-2: ESQUEMA REAL DE PROCESO DE TRABAJO

Como era de esperar, el proceso de datos está vigente a lo largo de todo el proyecto, ya que desde los primeros esbozos de la base de datos inicial, hasta la verdadera base de datos definitiva, el trabajo con los datos es continuo.

Es en el Desarrollo del sistema, dividido en su Base de datos y su plataforma, en lo que se centra este PFC.

### 4.1 Introducción, estructura del Siscue.

Al elegir un sistema de información como solución, se planteó el siguiente diagrama general de un sistema específico para los requerimientos del Vicerrectorado de Investigación de la UPM:

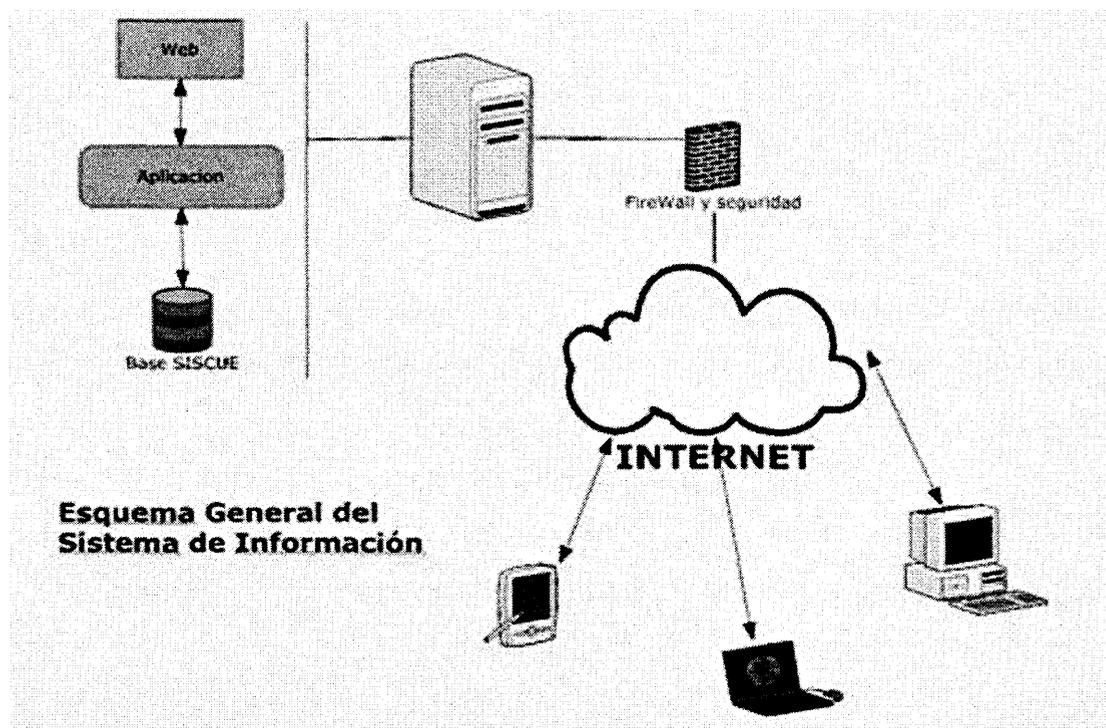


FIGURA 4-1: ESQUEMA GENERAL DEL SISTEMA DE INFORMACIÓN

Basándonos en este esquema, el verdadero desarrollo del sistema está en la parte instalada en el servidor, donde podemos entrar más a fondo en la siguiente figura:

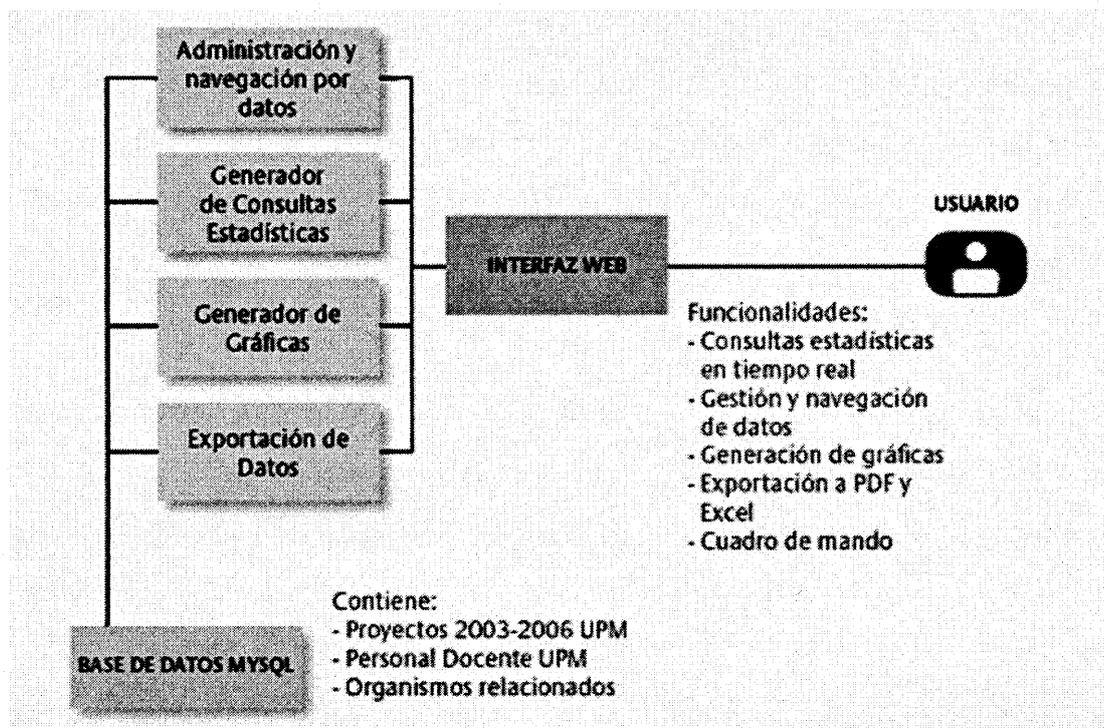


FIGURA 4-2: DIAGRAMA BASE DEL SISCUE

Con esta estructura básica se pretende abarcar toda la información necesaria para que el Vicerrectorado de Investigación se provea del análisis a lo largo del tiempo. Como se podrá ver esta estructura está en constante crecimiento, para contemplar todos los aspectos relacionados con la investigación.

En el esquema se diferencian rápidamente 2 bloques principales:

- Base de datos
- Plataforma, dividida en:
  - Aplicación intermedia para el manejo de la bdd.
  - Interfaz al usuario.

Uno de los objetivos de este PFC, es el diseño modular, con lo que cualquiera de estas 3 partes (base de datos, aplicación e interfaz) una vez diseñados, puede ser desarrollado independientemente si se acogen a lo fijado en ellos. Realmente se ha realizado por separado, pero junto en el tiempo, donde el crecimiento del Sistema ha afectado, obviamente, a las 3 partes del sistema.

**Base de datos:** El punto 4.2 reúne toda la información específica de la bdd, pero grosso modo se busca una base de datos relacional, multihilo y multiusuario, donde se puedan realizar a la vez gestión y consultas, de una manera centralizada. Es imprescindible que existan librerías para el acceso/manipulación a la tecnología que se elija.

**Se eligió:** → MySQL 5.0

**Aplicación:** Como se ve en detalle en el punto 4.3, consta de una aplicación dividida en paquetes para el manejo de toda la base de datos, y que a la vez provea a un interfaz Web.

**Se eligió:** → Java

**Interfaz Web:** Por su versatilidad, su sencillez, y su amplio abanico de opciones se eligió crear un interfaz web como punto de acceso. Gracias a la web 2.0, las ventajas de esta elección son numerosas, como la posibilidad del acceso remoto desde cualquier punto, la falta de requerimientos de software para que un usuario pueda usar el sistema, el amplio conocimiento de uso de los navegadores actuales por parte de casi todos los usuarios, la cantidad de utilidades, ampliaciones y demás que apoyan a los navegadores web actuales, que hacen de estos ya una herramienta muy completa para cualquier tipo de sistema, teniendo hasta aplicaciones multimedia que se basan en estas tecnologías.

**Se eligió:** → jsp, HTML y JavaScript.

## 4.2 Bases de datos.

Hay numerosas tecnologías de bases de datos muy completas para la realización de sistemas de información como el que se pretende, MySQL fue la elección debido a la compatibilidad que representa con las tecnologías que se querían aplicar y su licencia gratuita, una descripción de MySQL puede ser:

*“MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.*

*MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.”*

**Fuente:** <http://www.espestudio.com/a> , Especialistas web.

Una vez elegida esta tecnología, pasamos al diseño de nuestra base de datos, para ello tenemos que tener muy en cuenta tres factores:

- Que **abarque toda** la información que necesitemos ahora y en un futuro.
- Que sirva para la **gestión** de los datos.
- Que sea **relacional** y bien diseñada, para que la inserción y manipulación de datos sea lógica y correcta.
- Que sobre la base de datos podamos realizar **cualquier tipo de consulta** de la manera más óptima posible.

Para ello hay que tener en cuenta los elementos más importantes de información, que como se ha definido anteriormente son: Proyectos, Investigadores, Organismos y Grupos de Investigación. Sobre estos elementos hay que construir todas las relaciones que los unen.

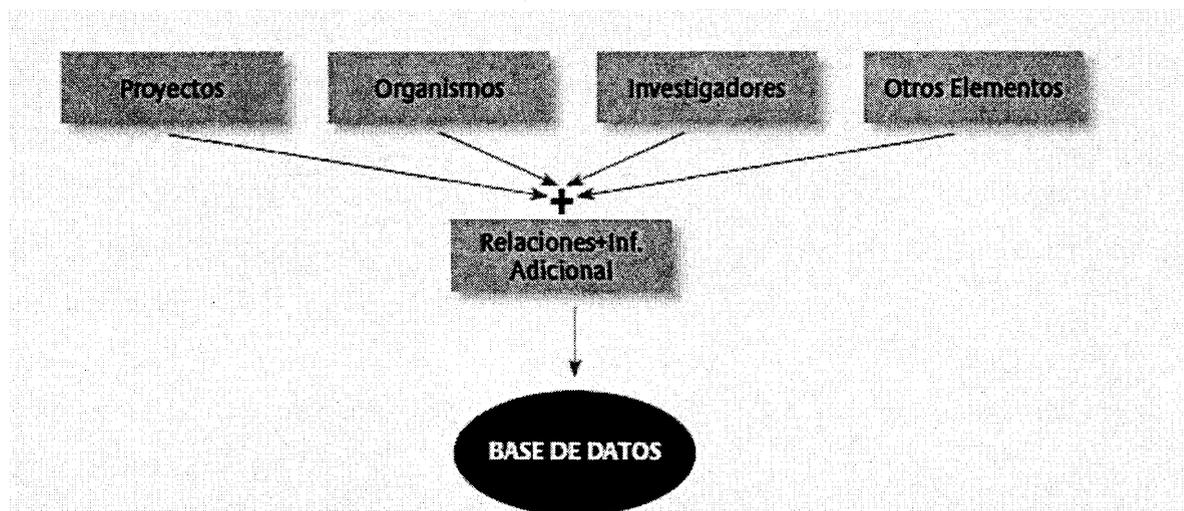


FIGURA 4-2: ELEMENTOS Y RELACIONES

Su diseño se puede dividir en:

- Diseño del modelo de datos:
  - Diseño del diagrama entidad-relación.
  - Creación del diagrama conceptual y físico basándonos en el diagrama anterior.
- Una vez el modelo de datos esté terminado, hay que elegir cuidadosamente toda la información normalizada que va a existir en el, como por ejemplo países, centros UPM, etc... Información que se intentará que sea fija y no modificable en el futuro.

### **Diseño del modelo de datos:**

Una vez identificados los elementos más importantes en los que se apoya nuestra base de datos, hay que decidir como van a relacionarse entre sí, por ejemplo, un centro está dividido en departamentos, donde éstos, solo pueden pertenecer a un centro. Un investigador puede pertenecer a un departamento (ergo a un centro) o no (por ejemplo, un contratado con cargo a proyecto). Un proyecto puede tener varios investigadores pero solo uno será el responsable de dicho proyecto, así mismo un proyecto puede tener un consorcio ilimitado de organismos relacionados, y un organismo puede estar relacionado con varios proyectos.

En cuanto a grupos de investigación, un grupo puede tener varios investigadores (de los que solo uno será el responsable) y un investigador solo puede pertenecer a la vez a un grupo de investigación. Para los centros y departamentos de los grupos de investigación se hace a través de sus componentes. Esto es solo parte de toda la lógica relacional que hay que tener en cuenta a la hora de “moldear” la base entera.

Teniendo todo esto en cuenta podemos crear el siguiente esquema de Entidad-Relación ya definitivo:

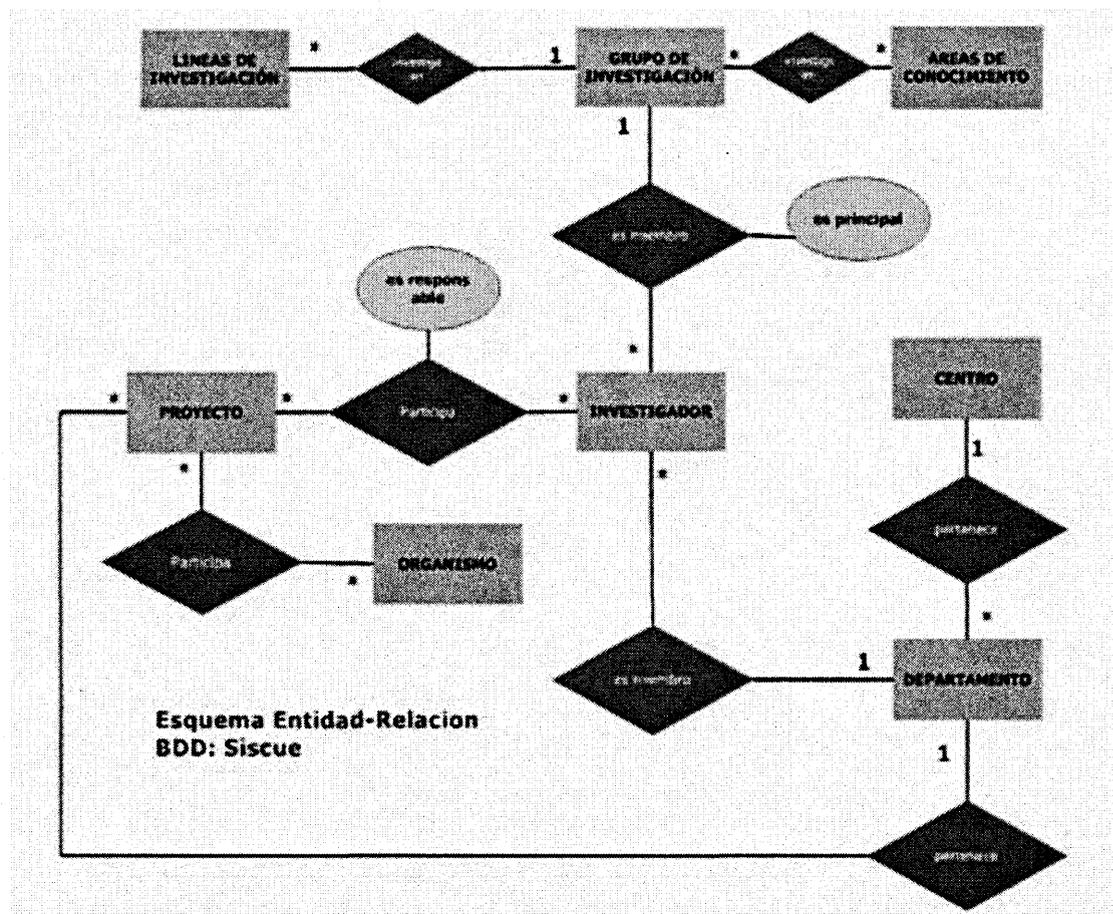


FIGURA 4-3: ESQUEMA ENTIDAD-RELACION BDD: SISCUE

En el diagrama de entidad-relación, se han omitido los atributos identificativos de los elementos debido a su complejidad, limitándonos a ver simplemente las relaciones que toman todos estos elementos. Además hay muchos elementos

que no se han incorporado en la parte de Grupos de Investigación ya que en la fecha de escritura de este libro está en fase de desarrollo.

Siendo fiel a la estructura entidad-relación vista, se elabora una base de datos ya completa con todas las tablas, con los datos identificativos de cada elemento y relación, y con todos los atributos necesarios para su posterior uso.

Para contemplar todas las necesidades, el diagrama de tablas de toda la base de datos queda así (hay que tener en cuenta que no se muestran todas las tablas, ya que se han omitido las usadas para gestión de la aplicación, y que en las tablas no se muestran los atributos ya que debido a la enormidad de ellos, hace más difícil una correcta visualización del diagrama):

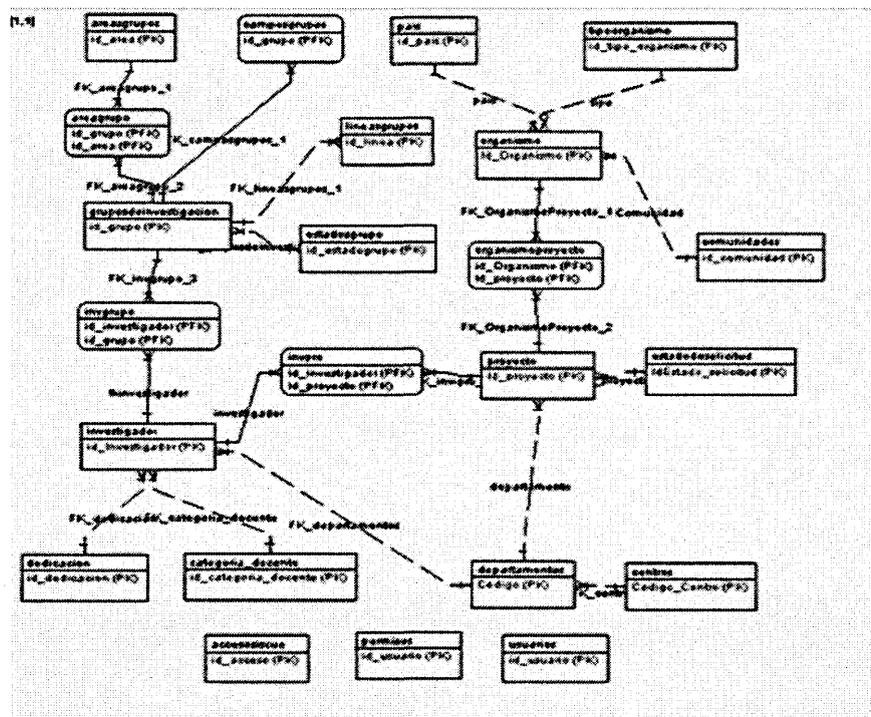


FIGURA 4-4: DIAGRAMA DE TABLAS DE LA BASE DE DATOS

Al ver este diagrama nos damos cuenta de que hay muchas más tablas de las que a priori se estimaban en el diagrama de Entidad-Relación (Sobre todo en grupos de investigación), esto es debido a dos factores:

1. Hay mucha información de carácter informativo que no nos condicionan la estructura inicial y en las tablas sí se puede distinguir.
2. Hay parte de la estructura que es "complementaria" a la inicial y fue ampliada después de implementar la inicial.

Todas estas tablas tienen los atributos que los describen, hemos omitido una definición completa de todas ellas debido a su magnitud, pero podemos ver a continuación un resumen de los elementos más importantes y los tipos de atributos que los completan.

• **Proyectos:**

<b>Campos</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
ID del proyecto	Entero	Un identificador que identifica unívocamente a cada proyecto
Campos descriptivos	Texto	Numerosos campos descriptivos e informativos sobre el proyecto, como el código OTT, título, observaciones, etc...
Campos de relación	Entero	Identificadores que relacionan al proyecto con otras partes de la base de datos para dar un resultado más completo sobre dicho proyecto, como son los identificadores de recursos económicos, de tipos de proyectos,
Fechas	Fecha	Fechas relacionadas con el proyecto, inicio, fin, entrada en la OTT del proyecto UPM, etc.

TABLA 4-1: DESCRIPCIÓN DE PROYECTO

• **Organismo:**

<b>Campos</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
ID del organismo	Entero	Un identificador que identifica unívocamente a cada organismo
Campos descriptivos	Texto	Numerosos campos descriptivos e informativos sobre el organismo, como el nombre, teléfonos, web, fax, etc..
Campos identificativos	Textp	En el caso del organismo tenemos el CIF que identifica de manera secundaria a cada organismo
Campos de relación	Entero	Igual que cualquier elemento de la base de datos para completarlo se tienen numerosas relaciones con el resto de elementos de la base de datos como países, comunidades, CNAE, tipo de organismo, etc

TABLA 4-2: DESCRIPCIÓN DE ORGANISMO

• **Investigador:**

<b>Campos</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
ID del investigador	Entero	Un identificador que identifica unívocamente a cada investigador
Campos descriptivos	Texto	Numerosos campos descriptivos e informativos sobre el investigador, teléfono, e-mail, etc.
Campos identificativos	Textp	DNI
Campos de relación	Entero	Como anteriormente el investigador se relaciona con Grupos de investigación, tipos de investigadores, departamento al que pertenece, etc.

TABLA 4-3: DESCRIPCIÓN DE INVESTIGADOR

• **Centros y Departamentos de la UPM:**

<b>Campos</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
Códigos	Texto	Códigos normalizados que identifican a los departamentos o centros, o entre ellos.
Campos descriptivos	Texto	Nombre del centro o departamento, etc.
Campos de relación	Texto	Relaciones entre ellos, (un departamento pertenece a un centro), etc.

TABLA 4-4: DESCRIPCIÓN DE CENTRO O DEPARTAMENTO

Lo anterior se completa con las tablas que llevan información de las relaciones entre todas ellas:

• **Relación Investigador – Proyecto**

<b>Campos</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
ID Proyecto	Entero	Identificador del proyecto
ID Investigador	Entero	Identificador del investigador
Es responsable	Booleano	Nos permite saber si en esta relación, el investigador es el responsable del proyecto

TABLA 4-5: DESCRIPCION DE RELACIONES INVESTIGADOR-PROYECTO

### Relación Organismo – Proyecto

Campos	TIPO	DESCRIPCION
ID Proyecto	Entero	Id del proyecto
ID Organismo	Entero	Id del organismo
Datos de contacto	Varios	Diferentes datos sobre el contacto de un organismo con un proyecto relacionado

TABLA 4-6: DESCRIPCIÓN DE RELACIONES ORGANISMO-PROYECTO

A estas relaciones se le suman muchísimas otras como investigadores con sus departamentos, organismos con sus tipos, proyectos con sus recursos económicos, etc.

Un ejemplo de una tabla de las anteriormente expuestas con todos sus atributos sería:

investigador			
id_investigador	Integer	NN	(PK)
Apellidos_investigadorIP	Varchar(250)		
Nombre_investigadorIP	Varchar(250)		
codigo_interno_investigadorIP	Varchar(250)		
DNI_investigador	Varchar(250)	NN	
Es_IP_investigadorIP	Bit(1)		
NRP_investigadorIP	Varchar(50)		
Email_investigadorIP	Varchar(250)		
Telefono_investigadorIP	Varchar(250)		
Fax_investigadorIP	Varchar(250)		
Id_departamento	Integer		
Id_grupo	Integer	NN	
Sexo_investigadorIP	Varchar(50)		
Edad_investigadorIP	Varchar(50)		
Pais_nacionalidad_investigadorIP	Varchar(50)		
Id_tipo_investigador	Integer	NN	
Dedicacion_al_proyecto_investigadorIP	Integer		
modo	Varchar(45)		
Codigo_Departamento	Varchar(45)	NN	
codigo_persona_temporal	Integer	NN	
id_dedicacion	Integer	NN	
Sexenios	Double		
Anio_tesis	Integer		
id_categoria_docente	Integer	NN	
Doctor	Tinyint		

FIGURA 4-5: TABLA DE INVESTIGADOR

Como se puede ver cada atributo tiene un nombre, un tipo, y unas propiedades. El nombre identifica el atributo, el tipo de dato indica que puede ser almacenado ahí, por ejemplo Varchar(45) se refiere a una cadena de caracteres de cómo máximo 45, Integer es un entero y bit(1) es un valor booleano. Se puede ver que una de las propiedades del valor "Id\_investigador" es PK (Primary key), que hace constar que es el valor que identifica **unívocamente** a cada elemento investigador, no pudiendo encontrarnos dos valores iguales en la tabla. Algunos de ellos son FK (Foreign key), que son identificadores de otros elementos de la base de datos (como un puntero que

apunta a ellos). Por ultimo las NN que aparecen significan "Not null", es un valor que no puede estar vacío.

Una vez completado todo lo anterior ya tendríamos completo el modelo de datos. Del cual nos ha quedado el siguiente resumen de tablas (sumando tanto las que vimos en el diagrama, como las tablas usadas para gestión de la aplicación), dividiéndolas en tablas usadas para la estructura básica, como puede ser la de investigador, y tablas secundarias usadas para, sobre todo, relaciones y propiedades de dichas relaciones.

<b>Tipo lógico de tablas</b>	<b>Cantidad</b>
Estructura básica	17
Secundarias	9
Total tablas	26

TABLA 4-7: RESUMEN DE TIPOS DE TABLAS

El Diagrama de clases final (resumido), sin tener en cuenta los grupos de investigación, con los atributos de todos los elementos es:

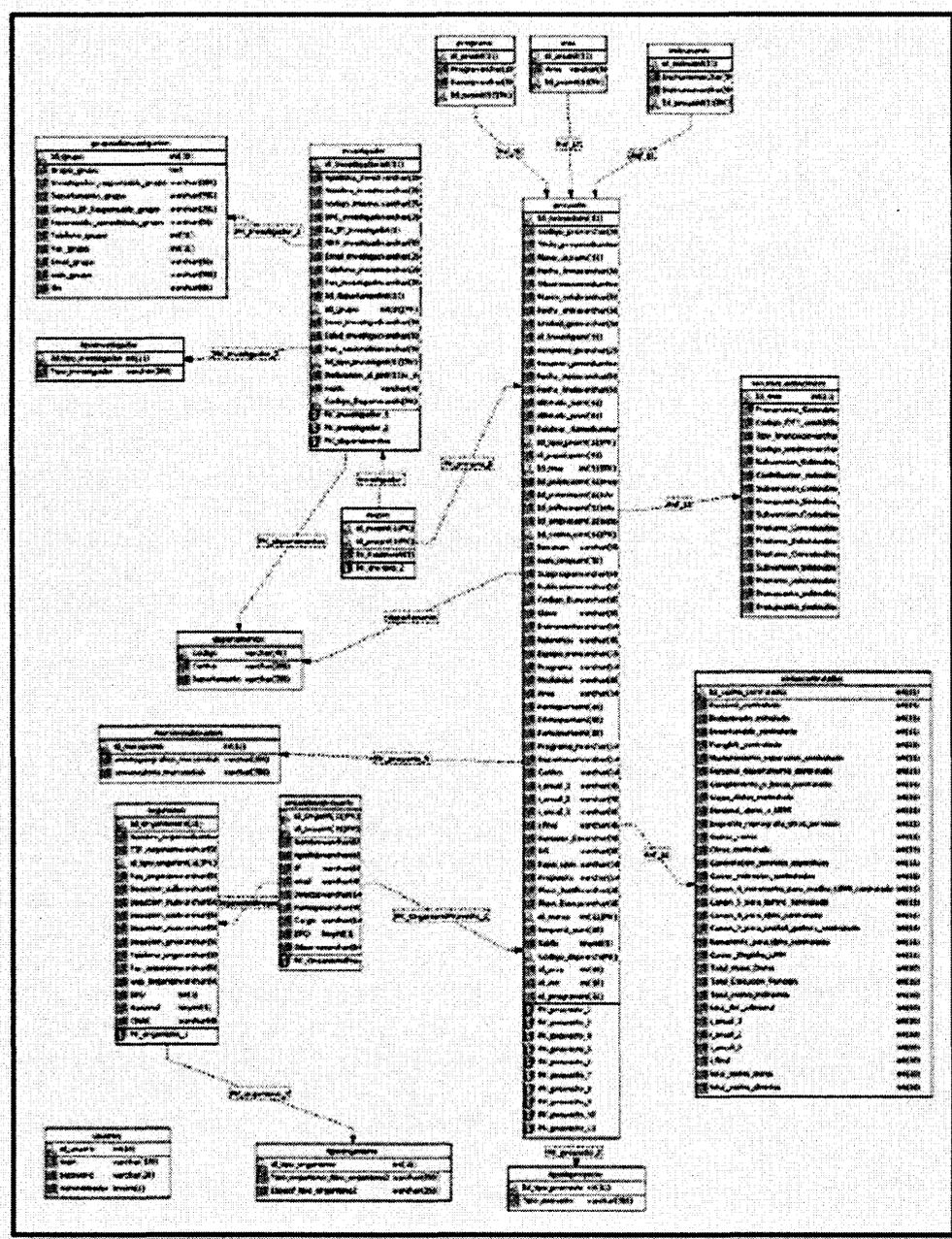


FIGURA 4-6: DIAGRAMA DE TABLAS FINAL

Usando software para el diseño de bases de datos (Power Designer) del modelo de datos completo, el script SQL que nos genera del modelo de datos (guardado en acuse.sql) es:

----- INICIO Script accuse.sql -----

```
/*!40101 SET NAMES utf8 */;
```

```
/*!40101 SET SQL_MODE=""*/;
```

```
create database if not exists `acuse`;
```

```
USE `acuse`;
```

```
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*Table structure for table `accesosiscue` */
```

```
DROP TABLE IF EXISTS `accesosiscue`;
```

```
CREATE TABLE `accesosiscue` (  
  `id_acceso` int(10) unsigned NOT NULL auto_increment,  
  `login` varchar(45) NOT NULL default "",  
  `pass` varchar(45) NOT NULL default "",  
  `fecha` datetime NOT NULL default '0000-00-00 00:00:00',  
  PRIMARY KEY (`id_acceso`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `actividad_linea` */
```

```
DROP TABLE IF EXISTS `actividad_linea`;
```

```
CREATE TABLE `actividad_linea` (  
  `id_actividad` int(10) unsigned NOT NULL auto_increment,  
  `id_linea` int(10) unsigned NOT NULL default '0',  
  `nombre_actividad` varchar(145) NOT NULL default "",  
  `descripcion_actividad` mediumtext NOT NULL,  
  `img_actividad` varchar(45) default NULL,  
  PRIMARY KEY (`id_actividad`),  
  KEY `FK_actividad_linea_1` (`id_linea`),  
  CONSTRAINT `FK_actividad_linea_1` FOREIGN KEY (`id_linea`) REFERENCES  
  `lineasgrupos` (`id_linea`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `actividad_proyecto` */
```

```
DROP TABLE IF EXISTS `actividad_proyecto`;
```

```
CREATE TABLE `actividad_proyecto` (  
  `id_proyecto` int(11) NOT NULL default '0',  
  `actividad_proyecto` varchar(245) default NULL,  
  `id_actividad_proyecto` int(10) unsigned NOT NULL auto_increment,  
  PRIMARY KEY (`id_actividad_proyecto`),  
  KEY `FK_actividad_proyecto_1` (`id_proyecto`),  
  CONSTRAINT `FK_actividad_proyecto_1` FOREIGN KEY (`id_proyecto`)  
  REFERENCES `proyecto` (`id_proyecto`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `area` */
```

```
DROP TABLE IF EXISTS `area`;
```

```
CREATE TABLE `area` (  
  `id_area` int(10) unsigned NOT NULL default '0',  
  `Area` varchar(75) default NULL,  
  PRIMARY KEY (`id_area`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `areagrupo` */
```

```
DROP TABLE IF EXISTS `areagrupo`;
```

```
CREATE TABLE `areagrupo` (  
  `id_grupo` varchar(8) NOT NULL default "",  
  `id_area` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_grupo`,`id_area`),  
  KEY `FK_areagrupo_1` (`id_area`),  
  CONSTRAINT `FK_areagrupo_1` FOREIGN KEY (`id_area`) REFERENCES  
  `areasgrupos` (`id_area`),  
  CONSTRAINT `FK_areagrupo_2` FOREIGN KEY (`id_grupo`) REFERENCES  
  `gruposdeinvestigacion` (`id_grupo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `areasgrupos` */
```

```
DROP TABLE IF EXISTS `areasgrupos`;
```

```
CREATE TABLE `areasgrupos` (  
  `id_area` int(10) unsigned NOT NULL default '0',  
  `area` varchar(255) default NULL,  
  PRIMARY KEY (`id_area`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `ayuda_pedida` */
```

```
DROP TABLE IF EXISTS `ayuda_pedida`;
```

```
CREATE TABLE `ayuda_pedida` (  
  `id_grupo` varchar(255) NOT NULL,  
  `anio` double NOT NULL default '0',  
  `concedida` tinyint(1) default NULL,  
  PRIMARY KEY (`id_grupo`,`anio`),  
  CONSTRAINT `FK_ayuda_pedida_1` FOREIGN KEY (`id_grupo`) REFERENCES  
  `gruposdeinvestigacion` (`id_grupo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `ayudagrupo` */
```

```
DROP TABLE IF EXISTS `ayudagrupo`;
```

```
CREATE TABLE `ayudagrupo` (  
  `Referencia` varchar(50) default NULL,  
  `Cantidad_concedida` double default NULL,
```

```
`Titulo_actividad` varchar(250) default NULL,  
`Num_registro` int(10) unsigned default NULL,  
`Area_ANEP` varchar(250) default NULL,  
`Tipo_ayuda` varchar(50) default NULL,  
`Palabra_clave` varchar(250) default NULL,  
`Apellidos_solicitante` varchar(250) default NULL,  
`Nombre_solicitante` varchar(250) default NULL,  
`Material_inv_solicitado` int(10) unsigned default NULL,  
`Gastos_ejecucion` int(10) unsigned default NULL,  
`Becas` int(10) unsigned default NULL,  
`Total_importe_solicitado` int(10) unsigned default NULL,  
`Observaciones_E` varchar(255) default NULL,  
`Puntuacion_UPM` int(10) unsigned default NULL,  
`Puntuacion_ANEP` int(10) unsigned default NULL,  
`Total_UPM_ANEP` int(10) unsigned default NULL,  
`Puntuacion_CM` int(10) unsigned default NULL,  
`Valoracion_total` int(10) unsigned default NULL,  
`Concedido_inv` double default NULL,  
`Concedido_gastos_ejecucion` double default NULL,  
`Concedido_becas` double default NULL,  
`id_ayuda` int(10) unsigned NOT NULL auto_increment,  
`id_grupo` varchar(15) default '0',  
`Fecha_creacion` datetime default NULL,  
`Fecha_modificacion` datetime default NULL,  
`Fecha Eliminacion` datetime default NULL,  
`Fecha` datetime default NULL,  
`Num_convocatoria` int(10) unsigned NOT NULL,  
PRIMARY KEY (`id_ayuda`),  
KEY `id_ayuda` (`id_ayuda`),  
KEY `id_grupo` (`id_grupo`),  
CONSTRAINT `FK_ayudagrupo_1` FOREIGN KEY (`id_grupo`) REFERENCES  
`gruposdeinvestigacion` (`id_grupo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
/*Table structure for table `camposgrupos` */  
  
DROP TABLE IF EXISTS `camposgrupos`;  
  
CREATE TABLE `camposgrupos` (  
  `id_grupo` varchar(8) NOT NULL default "",  
  `TAM` int(10) unsigned NOT NULL default '0',  
  `01` longtext,  
  `02` longtext,  
  `03` longtext,  
  `04` longtext,  
  `05` longtext,  
  `06` longtext,  
  `07` longtext,  
  `08` longtext,  
  `09` longtext,  
  PRIMARY KEY (`id_grupo`),  
  CONSTRAINT `FK_camposgrupos_1` FOREIGN KEY (`id_grupo`) REFERENCES  
  `gruposdeinvestigacion` (`id_grupo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

/*Table structure for table `categoria` */

DROP TABLE IF EXISTS `categoria`;

CREATE TABLE `categoria` (
  `categoria` varchar(255) default NULL,
  `tipo` varchar(10) default NULL,
  `regla` varchar(5) default NULL,
  `cod_categoria` int(10) unsigned NOT NULL default '0',
  `orden` int(10) unsigned default NULL,
  `campo_grupo` varchar(4) NOT NULL default 'S',
  PRIMARY KEY (`cod_categoria`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Table structure for table `categoria_docente` */

DROP TABLE IF EXISTS `categoria_docente`;

CREATE TABLE `categoria_docente` (
  `id_categoria_docente` int(10) unsigned NOT NULL default '0',
  `categoria_docente` varchar(45) NOT NULL default "",
  PRIMARY KEY (`id_categoria_docente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Table structure for table `categoriastamaño` */

DROP TABLE IF EXISTS `categoriastamaño`;

CREATE TABLE `categoriastamaño` (
  `id_categoriastam` int(10) unsigned NOT NULL auto_increment,
  `tipo` varchar(30) NOT NULL default "",
  `id_categoria` int(10) unsigned NOT NULL default '0',
  `doctor` tinyint(1) default NULL,
  `Descripcion` varchar(145) NOT NULL default "",
  PRIMARY KEY (`id_categoriastam`),
  KEY `FK_categoria` (`id_categoria`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Table structure for table `centros` */

DROP TABLE IF EXISTS `centros`;

CREATE TABLE `centros` (
  `Codigo_Centro` varchar(45) NOT NULL default "",
  `Centro` varchar(145) default NULL,
  `cod_memorias` char(2) NOT NULL default '00',
  `tipo` varchar(10) NOT NULL default 'centro',
  PRIMARY KEY (`Codigo_Centro`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Table structure for table `comunidades` */

DROP TABLE IF EXISTS `comunidades`;

CREATE TABLE `comunidades` (

```

```
`id_comunidad` int(10) unsigned NOT NULL auto_increment,  
`Comunidad` varchar(45) NOT NULL default "",  
PRIMARY KEY (`id_comunidad`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `costescontratados` */
```

```
DROP TABLE IF EXISTS `costescontratados`;
```

```
CREATE TABLE `costescontratados` (  
  `Id_costes_contratados` int(11) NOT NULL default '0',  
  `Personal_contratado` int(11) default NULL,  
  `Profesorado_cotrato` int(11) default NULL,  
  `Inventariable_contratado` int(11) default NULL,  
  `Fungible_contratado` int(11) default NULL,  
  `Mantenimiento_reparacion_contratado` int(11) default NULL,  
  `Personal_departamento_contratado` int(11) default NULL,  
  `Complemento_o_becas_contratado` int(11) default NULL,  
  `Viajes_dietas_contratado` int(11) default NULL,  
  `Personal_ajeno_a_UPM` int(11) default NULL,  
  `Imprenta_reprografia_otros_servicios` int(11) default NULL,  
  `Gastos_varios` int(11) default NULL,  
  `Otros_contratado` int(11) default NULL,  
  `Contratacion_servicios_especificos` int(11) default NULL,  
  `Costes_indirectos_contratados` int(11) default NULL,  
  `Canon_4_incremento_para_credito_UPM_contratado` int(11) default NULL,  
  `Canon_5_para_centro_contratado` int(11) default NULL,  
  `Canon_4_para_dpto_contratado` int(11) default NULL,  
  `Canon_2_para_unidad_gestora_contratado` int(11) default NULL,  
  `Remanente_para_dpto_contratado` int(11) default NULL,  
  `Costes_Elegibles_UPM` int(11) default NULL,  
  `Total_Viajes_Dietas` int(10) unsigned default NULL,  
  `Total_Ejecucion_Funcion` int(10) unsigned default NULL COMMENT 'total costes  
ejecucion o funcion',  
  `Total_costes_indirectos` int(10) unsigned default NULL,  
  `total_dot_adicional` int(10) unsigned default NULL,  
  `i_anual_1` int(10) unsigned default NULL,  
  `i_anual_2` int(10) unsigned default NULL,  
  `i_anual_3` int(10) unsigned default NULL,  
  `i_final` int(10) unsigned default NULL,  
  `total_costes_euros` int(10) unsigned default NULL,  
  `total_costes_directos` int(10) unsigned default NULL,  
  PRIMARY KEY (`Id_costes_contratados`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `dedicacion` */
```

```
DROP TABLE IF EXISTS `dedicacion`;
```

```
CREATE TABLE `dedicacion` (  
  `id_dedicacion` int(10) unsigned NOT NULL auto_increment,  
  `Dedicacion` varchar(45) NOT NULL default "",  
  PRIMARY KEY (`id_dedicacion`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `departamentos` */
```

```
DROP TABLE IF EXISTS `departamentos`;
```

```
CREATE TABLE `departamentos` (  
  `Codigo` varchar(45) NOT NULL default '0000',  
  `Centro` varchar(45) NOT NULL default "",  
  `Departamento` varchar(255) default NULL,  
  PRIMARY KEY (`Codigo`),  
  KEY `FK_centro` (`Centro`),  
  CONSTRAINT `FK_centro` FOREIGN KEY (`Centro`) REFERENCES `centros`  
  (`Codigo_Centro`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `estadosgrupo` */
```

```
DROP TABLE IF EXISTS `estadosgrupo`;
```

```
CREATE TABLE `estadosgrupo` (  
  `id_estadogrupos` int(10) unsigned NOT NULL auto_increment,  
  `estado` varchar(45) NOT NULL default "",  
  PRIMARY KEY (`id_estadogrupos`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `gruposdeinvestigacion` */
```

```
DROP TABLE IF EXISTS `gruposdeinvestigacion`;
```

```
CREATE TABLE `gruposdeinvestigacion` (  
  `id_grupo` varchar(8) NOT NULL default "",  
  `nombre` mediumtext,  
  `formacion` varchar(255) default NULL,  
  `ACRONIMO` varchar(50) default NULL,  
  `tlf` varchar(50) default NULL,  
  `FAX` varchar(50) default NULL,  
  `Web` varchar(250) default NULL,  
  `email` varchar(150) default NULL,  
  `Observaciones` varchar(255) default NULL,  
  `codigo_ott` varchar(50) default NULL,  
  `Convocatoria` enum('1a','2a','3a','4a','5a','6a','7a','8a','9a','10a') NOT NULL default '1a',  
  `id_estadogrupos` int(10) unsigned NOT NULL default '1',  
  `fecha_registro` date default NULL,  
  `num_registro` varchar(45) default NULL,  
  `img_grupo` varchar(45) default NULL,  
  `visible_web` tinyint(1) NOT NULL default '1',  
  PRIMARY KEY (`id_grupo`),  
  KEY `FK_gruposdeinvestigacion_1` (`id_estadogrupos`),  
  CONSTRAINT `FK_gruposdeinvestigacion_1` FOREIGN KEY (`id_estadogrupos`)  
  REFERENCES `estadosgrupo` (`id_estadogrupos`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `indicador` */
```

```
DROP TABLE IF EXISTS `indicador`;
```

```
CREATE TABLE `indicador` (  
  `id_indicador` int(10) unsigned NOT NULL default '0',  
  `id_primerero` int(10) unsigned NOT NULL default '0',  
  `DescripcionPrimero` varchar(145) NOT NULL default "",  
  `DescripcionSegundo` varchar(145) NOT NULL default "",  
  `peso_indicador` int(10) unsigned NOT NULL default '0',  
  `peso_grupo` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_indicador`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 9216 kB,  
peso total 27';
```

```
/*Table structure for table `indicadores_ver` */
```

```
DROP TABLE IF EXISTS `indicadores_ver`;
```

```
CREATE TABLE `indicadores_ver` (  
  `id_indicador_ver` int(10) unsigned NOT NULL auto_increment,  
  `grupo` varchar(145) NOT NULL default "",  
  `indicador` varchar(145) NOT NULL default "",  
  `nombre_campo` varchar(45) NOT NULL default "",  
  `orden_grupos` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_indicador_ver`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `instrumento` */
```

```
DROP TABLE IF EXISTS `instrumento`;
```

```
CREATE TABLE `instrumento` (  
  `id_instrumento` int(10) unsigned NOT NULL default '0',  
  `Instrumento_nombre` varchar(50) default NULL,  
  `Instrumento_acronimo` varchar(50) default NULL,  
  `Ambito_geo` varchar(250) NOT NULL default 'sin Marco',  
  PRIMARY KEY (`id_instrumento`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 10240 kB;  
(`id_marco`) REFER `acuse/marcocolabo`;
```

```
/*Table structure for table `investigador` */
```

```
DROP TABLE IF EXISTS `investigador`;
```

```
CREATE TABLE `investigador` (  
  `Apellidos_investigadorIP` varchar(250) default NULL,  
  `Nombre_investigadorIP` varchar(250) default NULL,  
  `codigo_interno_investigadorIP` varchar(250) default NULL,  
  `DNI_investigador` varchar(250) NOT NULL default "",  
  `Es_IP_investigadorIP` bit(1) default NULL,  
  `NRP_investigadorIP` varchar(50) default NULL,  
  `Email_investigadorIP` varchar(250) default NULL,  
  `Telefono_investigadorIP` varchar(250) default NULL,  
  `Fax_investigadorIP` varchar(250) default NULL,  
  `Id_departamento` int(11) default NULL,  
  `Id_grupo` int(10) unsigned NOT NULL default '0',  
  `Sexo_investigadorIP` varchar(50) default NULL,  
  `Edad_investigadorIP` varchar(50) default NULL,
```

```

`Pais_nacionalidad_investigadorIP` varchar(50) default NULL,
`Id_tipo_investigador` int(11) NOT NULL default '0',
`Dedicacion_al_proyecto_investigadorIP` int(11) default NULL,
`id_Investigador` int(11) NOT NULL default '0',
`modo` varchar(45) default NULL,
`Codigo_Departamento` varchar(45) NOT NULL default '0000',
`codigo_persona_temporal` int(10) unsigned NOT NULL default '0',
`id_dedicacion` int(10) unsigned NOT NULL default '1',
`Sexenios` double default NULL,
`Anio_tesis` int(4) unsigned default NULL,
`Doctor` tinyint(1) default NULL,
`cod_categoria` int(10) unsigned NOT NULL default '0',
`contrato` varchar(145) default NULL,
`id_categoria_docente` int(10) unsigned default NULL,
`Cod_centro` varchar(2) NOT NULL default '00',
PRIMARY KEY (`id_Investigador`),
KEY `FK_investigador_1` (`Id_grupo`),
KEY `FK_investigador_2` (`Id_tipo_investigador`),
KEY `FK_departamentos` (`Codigo_Departamento`),
KEY `FK_dedicacion` (`id_dedicacion`),
KEY `categoriainv` (`cod_categoria`),
KEY `FK_investigador_centro` (`Cod_centro`),
CONSTRAINT `categoriainv` FOREIGN KEY (`cod_categoria`) REFERENCES
`categoria` (`cod_categoria`),
CONSTRAINT `FK_dedicacion` FOREIGN KEY (`id_dedicacion`) REFERENCES
`dedicacion` (`id_dedicacion`),
CONSTRAINT `FK_departamentos` FOREIGN KEY (`Codigo_Departamento`)
REFERENCES `departamentos` (`Codigo`),
CONSTRAINT `FK_investigador_centro` FOREIGN KEY (`Cod_centro`)
REFERENCES `centros` (`Codigo_Centro`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 14336 kB;
(`Id_grupo`) REFER `acuse/gruposdeinv`;

```

```

/*Table structure for table `invgrupo` */

```

```

DROP TABLE IF EXISTS `invgrupo`;

```

```

CREATE TABLE `invgrupo` (
  `id_investigador` int(11) NOT NULL auto_increment,
  `id_grupo` varchar(8) NOT NULL default '0',
  `es_responsable` tinyint(1) NOT NULL default '0',
  `Activo` tinyint(1) NOT NULL default '1',
  `Fecha_Alta` date default NULL,
  `Fecha_Baja` date default NULL,
  `Fecha_Modificacion` date default NULL,
  `id_area` int(11) NOT NULL default '0',
  `doctor` tinyint(1) unsigned default NULL,
  `Fecha_Alta_Sistema` date default NULL,
  `Fecha_Baja_Sistema` date default NULL,
  PRIMARY KEY (`id_investigador`,`id_grupo`),
  KEY `fkgrupo` (`id_grupo`),
  KEY `FK_area` (`id_area`),
  CONSTRAINT `fkinvestigador` FOREIGN KEY (`id_investigador`) REFERENCES
`investigador` (`id_Investigador`),

```

```
CONSTRAINT `FK_invgrupo_2` FOREIGN KEY (`id_grupo`) REFERENCES
`gruposdeinvestigacion` (`id_grupo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 52224 kB;
InnoDB free: 52224 kB; (`id_investiga`;
```

```
/*Table structure for table `invpro` */
```

```
DROP TABLE IF EXISTS `invpro`;
```

```
CREATE TABLE `invpro` (
  `id_investigador` int(11) NOT NULL default '0',
  `id_proyecto` int(11) NOT NULL default '0',
  `Es_Responsable` tinyint(1) NOT NULL default '0',
  PRIMARY KEY (`id_investigador`,`id_proyecto`),
  KEY `FK_inv-pro_2` (`id_proyecto`),
  CONSTRAINT `FK_inv-pro_2` FOREIGN KEY (`id_proyecto`) REFERENCES
`proyecto` (`id_proyecto`),
  CONSTRAINT `investigador` FOREIGN KEY (`id_investigador`) REFERENCES
`investigador` (`id_Investigador`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `lineasgrupos` */
```

```
DROP TABLE IF EXISTS `lineasgrupos`;
```

```
CREATE TABLE `lineasgrupos` (
  `id_grupo` varchar(255) NOT NULL,
  `id_linea` int(10) unsigned NOT NULL auto_increment,
  `linea` varchar(255) default NULL,
  `id_investigador` bigint(20) default NULL,
  `Descripcion` mediumtext,
  `nombre_imagen` varchar(45) default NULL,
  PRIMARY KEY (`id_linea`),
  KEY `FK_lineasgrupos_1` (`id_grupo`),
  CONSTRAINT `FK_lineasgrupos_1` FOREIGN KEY (`id_grupo`) REFERENCES
`gruposdeinvestigacion` (`id_grupo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `lineashojasextra` */
```

```
DROP TABLE IF EXISTS `lineashojasextra`;
```

```
CREATE TABLE `lineashojasextra` (
  `id_linea` int(10) unsigned NOT NULL default '0',
  `hojaextra` varchar(145) NOT NULL default "",
  PRIMARY KEY (`id_linea`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `marcocolaboracion` */
```

```
DROP TABLE IF EXISTS `marcocolaboracion`;
```

```
CREATE TABLE `marcocolaboracion` (
  `id_marcocolab` int(11) NOT NULL default '0',
  `ambitogeografico_marcocolab` varchar(250) NOT NULL default 'sin Marco',
```

```

`convocatoria_marcocolab` varchar(250) default NULL,
`marco_colaboracion_nivel2` varchar(145) default NULL,
PRIMARY KEY (`id_marcocolab`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Table structure for table `organismo` */

DROP TABLE IF EXISTS `organismo`;

CREATE TABLE `organismo` (
  `Id_Organismo` int(11) NOT NULL,
  `Nombre_organismo` mediumtext,
  `CIF_organismo` varchar(50) default NULL,
  `id_tipo_organismo` int(11) default NULL,
  `Direccion_calle_organismo` varchar(400) default NULL,
  `Direccion_codigo_postal_organismo` varchar(50) default NULL,
  `Direccion_ciudad_organismo` varchar(50) default NULL,
  `Direccion_provincia_organismo` varchar(50) default NULL,
  `Telefono_organismo` varchar(150) default NULL,
  `Fax_organismo` varchar(50) default NULL,
  `web_organismo` varchar(350) default NULL,
  `EPO` bit(1) default NULL,
  `Nacional` tinyint(1) default NULL,
  `CNAE` varchar(145) NOT NULL default "",
  `id_comunidad` int(10) unsigned NOT NULL default '1',
  `email` varchar(145) default NULL,
  `id_pais` int(10) unsigned NOT NULL default '0',
  `annocreacion` smallint(4) unsigned default NULL,
  `numempleados` smallint(6) unsigned default NULL,
  PRIMARY KEY (`Id_Organismo`),
  KEY `FK_organismo_1` (`id_tipo_organismo`),
  KEY `FK_organismo_2` (`id_comunidad`),
  KEY `pais` (`id_pais`),
  CONSTRAINT `Comunidad` FOREIGN KEY (`id_comunidad`) REFERENCES
`comunidades` (`id_comunidad`),
  CONSTRAINT `pais` FOREIGN KEY (`id_pais`) REFERENCES `pais` (`id_pais`),
  CONSTRAINT `tipo` FOREIGN KEY (`id_tipo_organismo`) REFERENCES
`tipoororganismo` (`id_tipo_organismo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 72704 kB;
(`id_tipo_organismo`) REFER `acuse/ti';

/*Table structure for table `organismoproyecto` */

DROP TABLE IF EXISTS `organismoproyecto`;

CREATE TABLE `organismoproyecto` (
  `id_Organismo` int(11) NOT NULL default '0',
  `id_proyecto` int(11) NOT NULL default '0',
  `Nombre_Contacto` varchar(65) default NULL,
  `Apellidos_Contacto` varchar(65) default NULL,
  `tlf` varchar(65) default NULL,
  `email` varchar(145) default NULL,
  `Direccion` varchar(145) default NULL,
  `Participacion` varchar(45) default NULL,
  `Cargo` varchar(145) default NULL,

```

```
`EPO` tinyint(1) NOT NULL default '0',  
`Observaciones` varchar(345) default NULL,  
PRIMARY KEY (`id_Organismo`,`id_proyecto`),  
KEY `FK_OrganismoProyecto_2` (`id_proyecto`),  
CONSTRAINT `FK_OrganismoProyecto_1` FOREIGN KEY (`id_Organismo`)  
REFERENCES `organismo` (`Id_Organismo`),  
CONSTRAINT `FK_OrganismoProyecto_2` FOREIGN KEY (`id_proyecto`)  
REFERENCES `proyecto` (`Id_proyecto`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `pais` */
```

```
DROP TABLE IF EXISTS `pais`;
```

```
CREATE TABLE `pais` (  
  `id_pais` int(10) unsigned NOT NULL default '0',  
  `pais` varchar(45) NOT NULL default "",  
  PRIMARY KEY (`id_pais`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `permisos` */
```

```
DROP TABLE IF EXISTS `permisos`;
```

```
CREATE TABLE `permisos` (  
  `id_usuario` int(10) unsigned NOT NULL default '0',  
  `Nacionales` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `Internacionales` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `Otros_Marcos` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `Organismos` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `personal_docente` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `personal_no_docente` enum('ver','administrar','nada','usuario') NOT NULL default  
'ver',  
  `Grupos` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `Plancalidad` enum('ver','administrar','nada','usuario') NOT NULL default 'ver',  
  `Estadisticas` enum('ver','administrar','nada','usuario') NOT NULL default 'nada',  
  `perfil` enum('ver','administrar','nada') NOT NULL default 'administrar',  
  PRIMARY KEY (`id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `programa` */
```

```
DROP TABLE IF EXISTS `programa`;
```

```
CREATE TABLE `programa` (  
  `id_programa` int(10) unsigned NOT NULL default '0',  
  `Programa` varchar(255) default NULL,  
  `Descripcion` varchar(50) default NULL,  
  `Ambito_Geo` varchar(250) NOT NULL default 'sin Marco',  
  PRIMARY KEY (`id_programa`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `proyecto` */
```

```
DROP TABLE IF EXISTS `proyecto`;
```

```

CREATE TABLE `proyecto` (
  `Id_proyecto` int(11) NOT NULL,
  `Codigo_proyecto` varchar(50) default NULL,
  `Titulo_proyecto` mediumtext,
  `Meses_duracion` int(11) NOT NULL default '0',
  `Fecha_firma_proyecto` varchar(50) default NULL,
  `Observaciones_proyecto` mediumtext,
  `Fecha_entrada_OTT_proyecto` varchar(50) default NULL,
  `Acronimo_proyecto` varchar(250) default NULL,
  `Resumen_proyecto` mediumtext,
  `Fecha_inicio_proyecto` varchar(50) default NULL,
  `Fecha_finalizacion_proyecto` varchar(50) default NULL,
  `Palabras_clave_proyecto` mediumtext,
  `Id_tipo_proyecto` int(11) NOT NULL default '0',
  `Id_rree` int(11) NOT NULL default '0',
  `Id_publicaciones_proyecto` int(11) default NULL,
  `Id_patentes_proyecto` int(11) default NULL,
  `Id_software_proyecto` int(11) default NULL,
  `Id_empresas_proyecto` int(11) default NULL,
  `Id_costes_contratados` int(11) default NULL,
  `Subprograma` varchar(65) default NULL,
  `Codigo_Europeos` varchar(45) NOT NULL default '0',
  `Clave` varchar(45) default NULL,
  `Referencia` varchar(245) default NULL,
  `Modalidad` varchar(45) default NULL,
  `Departamento` varchar(145) default NULL,
  `Centro` varchar(145) default NULL,
  `Europeo_Estado` varchar(45) default NULL,
  `Papel_upm` varchar(145) default NULL,
  `Propuesta` varchar(145) default NULL,
  `id_marco` int(11) NOT NULL default '0',
  `Valido` tinyint(1) NOT NULL default '0',
  `Codigo_departamento` varchar(45) NOT NULL default '0000',
  `id_area` int(10) unsigned NOT NULL default '0',
  `id_inst` int(10) unsigned NOT NULL default '0',
  `id_programa` int(10) unsigned NOT NULL default '0',
  `Anio` varchar(10) NOT NULL default 'vacio',
  `Organismo_Financiadore` int(11) NOT NULL default '0',
  `Codigo_Centro` varchar(45) default NULL,
  `Actividades` varchar(650) NOT NULL default "",
  `web` varchar(145) NOT NULL default "",
  `email` varchar(145) NOT NULL default "",
  `Creditos` int(10) unsigned NOT NULL default '0',
  `plazas` varchar(45) NOT NULL default 'Indefinido',
  PRIMARY KEY (`Id_proyecto`),
  KEY `centro` (`Codigo_Centro`),
  KEY `tipoProyecto` (`Id_tipo_proyecto`),
  KEY `RecursosEconomicos` (`Id_rree`),
  KEY `costes` (`Id_costes_contratados`),
  KEY `MarcoColaboracion` (`id_marco`),
  KEY `Departamento` (`Codigo_departamento`),
  KEY `area_proyecto` (`id_area`),
  CONSTRAINT `area_proyecto` FOREIGN KEY (`id_area`) REFERENCES `area`
(`id_area`),

```

```
CONSTRAINT `centro` FOREIGN KEY (`Codigo_Centro`) REFERENCES `centros`
(`Codigo_Centro`),
CONSTRAINT `departamento` FOREIGN KEY (`Codigo_departamento`)
REFERENCES `departamentos` (`Codigo`),
CONSTRAINT `MarcoColaboracion` FOREIGN KEY (`id_marco`) REFERENCES
`marcocolaboracion` (`id_marcocolab`),
CONSTRAINT `RecursosEconomicos` FOREIGN KEY (`Id_rree`) REFERENCES
`recursos_economicos` (`Id_rree`),
CONSTRAINT `tipoProyecto` FOREIGN KEY (`Id_tipo_proyecto`) REFERENCES
`tipodeproyecto` (`Id_tipo_proyecto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 72704 kB;
(`Codigo_Centro`) REFER `acuse/centro`;
```

```
/*Table structure for table `recursos_economicos` */
```

```
DROP TABLE IF EXISTS `recursos_economicos`;
```

```
CREATE TABLE `recursos_economicos` (
  `Id_rree` int(11) NOT NULL default '0',
  `Presupuesto_Concedido_UPM` double default NULL,
  `Codigo_OTT_organismo_financiator_rree` int(10) unsigned default NULL,
  `Tipo_financiacion_rree` varchar(50) default NULL,
  `Codigo_creditos_NABS_rree` varchar(50) default NULL,
  `Subvencion_Solicitada_UPM` double default NULL,
  `Contribucion_maxima_ce` double default NULL,
  `matricula_master` double NOT NULL default '0',
  PRIMARY KEY (`Id_rree`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `tipodeproyecto` */
```

```
DROP TABLE IF EXISTS `tipodeproyecto`;
```

```
CREATE TABLE `tipodeproyecto` (
  `Id_tipo_proyecto` int(11) NOT NULL default '0',
  `Tipo_proyecto` varchar(250) default NULL,
  PRIMARY KEY (`Id_tipo_proyecto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `tipoinvestigador` */
```

```
DROP TABLE IF EXISTS `tipoinvestigador`;
```

```
CREATE TABLE `tipoinvestigador` (
  `Id_tipo_investigador` int(11) NOT NULL default '0',
  `Tipo_investigador` varchar(250) default NULL,
  PRIMARY KEY (`Id_tipo_investigador`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `tipoorganismo` */
```

```
DROP TABLE IF EXISTS `tipoorganismo`;
```

```
CREATE TABLE `tipoorganismo` (
  `id_tipo_organismo` int(11) NOT NULL default '0',
```

```
`Tipo_organismo_tipo_organismo2` varchar(45) default NULL,  
`Especif_tipo_organismo2` varchar(145) default NULL,  
PRIMARY KEY (`id_tipo_organismo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
/*Table structure for table `usuarios` */  
  
DROP TABLE IF EXISTS `usuarios`;  
  
CREATE TABLE `usuarios` (  
  `id_usuario` int(10) unsigned NOT NULL auto_increment,  
  `login` varchar(120) NOT NULL default "",  
  `password` varchar(20) NOT NULL,  
  `Administrador` tinyint(1) NOT NULL default '0',  
  `nivel` int(10) unsigned NOT NULL default '0' COMMENT '2 proyectos, 3 grupos inv, ',  
  `descripcion` varchar(145) default NULL,  
  `nombre_completo` varchar(100) default NULL,  
  PRIMARY KEY (`id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
```

----- FIN Script acuse.sql -----

**Selección de los datos iniciales y normalizados para nuestra base:**

El siguiente paso es rellenar los datos normalizados que van a ser usados por el Sistema. Es muy importante la decisión de estos datos para el posterior uso y diseño de la aplicación, ya que condicionará en cierta medida las posibilidades de la misma.

A continuación se muestra un ejemplo de los tipos de organismos normalizados:

<b>Empresa</b>	<b>Universidad</b>
Microempresa: (<10)	Universidad Pública
Pequeña Empresa (11-50)	Universidad Privada
Mediana Empresa (51-250)	Grupo de Investigación
Gran Empresa (>250)	Centro Universitario de I+D
Persona Física	Instituto Universitario de Investigación
Centro de Innovación Empresarial	Fundación Universitaria
Asociación Empresarial	Otros Universidad
Fundación Empresarial	
Otros Empresa	

TABLA 4-8: TIPOS NORMALIZADOS DE ORGANISMO

<b>Administración</b>	<b>Entidad I+D+i</b>
Administración Europea	Organismo Público de Investigación
Administración Central	Centro de Innovación y Tecnología (Registrado en el MEC <sup>99</sup> )
Administración Autonómica	Otros Entidad de I+D+i
Administración Local	
Otros Administración	

TABLA 4-9: TIPOS NORMALIZADOS DE ORGANISMO

Otros
Laboratorio, Centro de Certificación y Organismo afin
Colegio o Consejo Profesional
Hospital
Entidad Deportiva
Comunidad de Propietarios
Otra Institución Privada sin Fines Lucrativos (ISPFL)
Organismo Internacional
Otros

TABLA 4-10: TIOS NORMALIZADOS DE ORGANISMO

Aparte de los tipos de organismos, hay muchos otros datos susceptibles a ser normalizados, que son:

- Areas temáticas
- Centros UPM
- Departamentos UPM
- Tipos de proyectos
- Categorías de un investigador
- Comunidades
- Países
- Estados de un proyecto

Una vez rellenos todos los datos normalizados de la base, está lista para empezar a usarse y rellenarse.

### **4.3 Plataforma**

Una vez construida la base de datos, es necesario una plataforma que nos permita no solo la manipulación de estos datos, sino la elaboración de resultados más complejos a partir de ellos. A todo esto hay que sumarle que el interfaz al usuario debe ser amigable, de tal forma que no le cueste ningún esfuerzo realizar toda la gestión y el análisis de todos los datos, por muy complejo que sea por debajo.

#### **4.3.1 Aplicación**

Entre la base de datos y el interfaz al usuario, corre software que hace de nivel intermedio. Los servicios predefinidos que tenía que abarcar esta aplicación son:

- **Navegación por los datos:** De tal forma que se pueda navegar con permisos de usuario (más o menos restringido) por todos los datos.
- **Administración de los datos:** Sobre todo modificación, agregado y eliminado de datos.
- **Generación de consultas** configurables por el usuario, de manera que pueda elaborar consultas muy complejas parametrizándolas para facilitarlas en gran medida.
- **Generación de gráficas:** Una vez realizadas las consultas, elaborar un sistema de creación de gráficas para diferentes tipos de consultas, de tal forma que se pueda elegir qué mostrar para su mejor entendimiento.
- **Exportación de datos:** Permitir que los datos, o las consultas, sean exportados a documentos con el formato de software más común con el usuario tipo (pdf para presentaciones o documentos de texto, Excel para consultas estadísticas, etc).

Para la realización de la aplicación se ha usado Java como lenguaje de programación de alto nivel. Aparte de todo el código elaborado para Siscue con las librerías comunes de java (java.util, java.lang, etc.), se ha importado y/o manipulado de ciertas librerías no comunes de código abierto.

Las librerías no comunes importadas (que serán descritas en los módulos que las usan) son:

- **POI:** Para la compatibilidad con los formatos Microsoft.
- **JfreeChart:** Para la generación de gráficas.
- **ODBC:** Software y librerías para la compatibilidad Java y My-SQL

Descripciones detalladas de las librerías POI y JfreeChart:

### **1. Librería para la compatibilidad con formato Microsoft.**

**Nombre:** POI

**Fuente:** Jakarta

**URL:** <http://poi.apache.org/>

**Descripción:** para una óptima descripción de esta librería se ha usado la suya propia, sacada de <http://poi.apache.org/>:

*“El proyecto POI consiste en APIs para manipular varios formatos de ficheros basados en el formato de Documento Compuesto OLE 2 de Microsoft, utilizando Java puro. En concreto, se pueden leer y escribir ficheros MS Excel utilizando Java. Pronto se podrá leer y escribir ficheros Word utilizando Java. POI es su solución Java Excel así como su solución Java Word. En cualquier caso, tenemos un API completo para portar otros formatos de Documento Compuesto OLE 2 y todo aquel que quiera participar será bienvenido.*

*Entre los ficheros basados en el formato de Documento Compuesto OLE 2 de Microsoft se incluyen la mayor parte de los ficheros de Microsoft Office tales como XLS y DOC así como formatos de fichero basados en el API de serialización MFC. “*

## 2. Librería para la generación de gráficas

**Nombre:** JFreeChart

**Fuente:** Object Refinery Limited

**URL:** <http://www.jfree.org/jfreechart/>

**Descripción:** Al igual que en la anterior librería, referenciamos una descripción hecha por los creadores:

*"JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:*

- *a consistent and well-documented API, supporting a wide range of chart types;*
- *a flexible design that is easy to extend, and targets both server-side and client-side applications;*
- *support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);*
- *JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications. "*

Cabe destacar que ambas son "Open Source" lo que ha permitido la manipulación de todo este software para adaptarlo a nuestras necesidades. Si se desea más información sobre ellas en las fuentes proporcionan todo el código y los Java-docs.

### 4.3.1.1 Diagrama general de la aplicación.

La aplicación se divide en paquetes que abarcan diferentes tipos de servicios, intentando modular toda la aplicación de tal forma que los paquetes suplan unos servicios específicos.

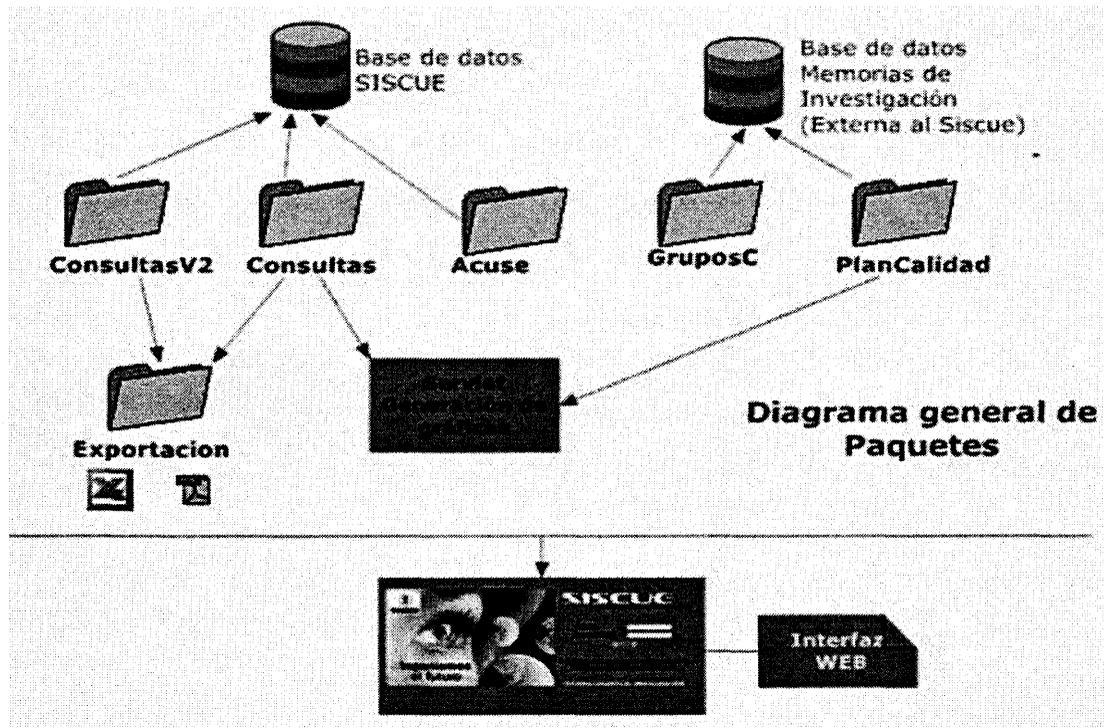


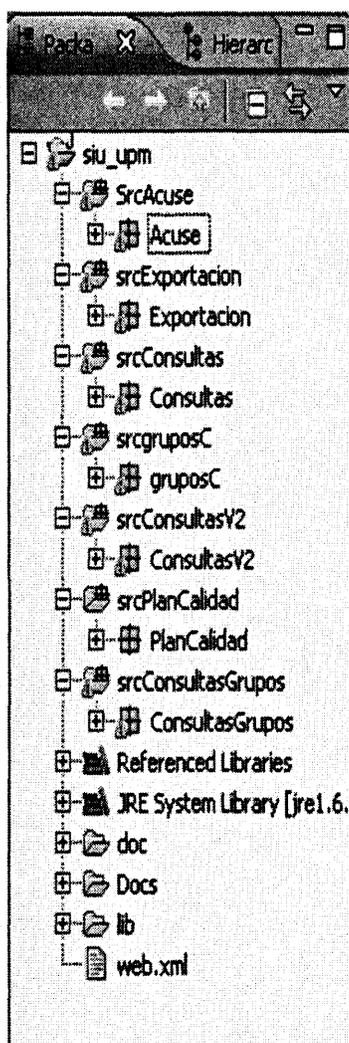
FIGURA 4-7: DIAGRAMA DE PAQUETES DE LA PLATAFORMA

Viendo el diagrama de paquetes, podemos diferenciar los siguientes:

- **Acuse:** Encargado de la navegación básica y la administración de la base de datos de SISCUE
- **Consultas:** Módulo encargado de la generación y normalización de cualquier tipo de consulta sencilla sobre proyectos en torno a otro tipo de datos (organismos, investigadores, centros, departamentos, países, programas europeos y grupos de investigación). El objetivo de este modulo es tener una estructura común a casi todos los tipos de datos para ser ampliamente usado en estadísticas sencillas.
- **ConsultasV2:** Consultas complejas especificando cada tipo de elemento sobre el que se realiza consultas, generando tablas configurables con

todos los datos requeridos. Tiene implementado dos elementos: Proyectos y Grupos de Investigación, aunque está preparada para ampliarse con más.

- **GruposC:** Encargado de la navegación básica y la administración de una base de datos externa de Memorias de Investigación (Para el plan de Calidad).
- **Exportacion:** Encargado de exportar tablas sacadas de consultas a Excel, o gráficas a pdf.
- **PlanCalidad:** Realiza todo el plan de calidad, cooperando con GruposC.
- **Servlet: GeneracionGraficasSiscue:** Servlet que recibiendo objetos Consulta genera graficas configurables.



Se ha usado eclipse como editor en el entorno de desarrollo, donde podemos ver la estructura de paquetes que se ajusta a los módulos diseñados.

Cada módulo tiene una clase principal que maneja todos los servicios que ofrece, usando numerosas estructuras para ello. En dichas clases siempre existe una método conector para la conexión con las bases de datos existentes en el Siscue, esto es porque cada clase tiene un tipo de conexión diferente (diferentes permisos en el schema, etc.) El inconveniente es que si el schema de las bases de datos o las condiciones de la conexión (puerto, tipo de conexión, etc.) cambian, hay que modificar el tipo de conexión en todos los paquetes.

#### 4.3.1.2 Descripción de los módulos.

Todos los módulos que tienen relación directa tienen un objeto conector que usando ODBC conecta con la Base de datos, el código de dicho conector usando el driver es (ejemplo de Acuse.java):

Método para conectar a la BDD, donde los parámetros url, login y password se le pasan a la clase principal convirtiéndose en atributos:

```
/**Conecta con la base de datos Acuse de LocalHost (Modificable en el
*código fuente por motivos de seguridad)
*@return "Conectado" si conecta, excepción descrita en caso de algún
*problema
**/
    static String bd="acuse ";
    static String login="prueba";
    static String password="prueba";
    static String url = "jdbc:mysql://localhost:3306/"+bd";
public String conectar() {

    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn=DriverManager.getConnection(url,login,password);
        return("Conectado");

    }catch (SQLException a){
        return(a.getMessage());

    }

}
} //de conectar()
```

### 4.3.1.2.1 Módulo Acuse

Encargado de la navegación básica y la administración de la base de datos de Siscue. Es con diferencia el modulo más amplio en código ya que maneja todas las estructuras del sistema.

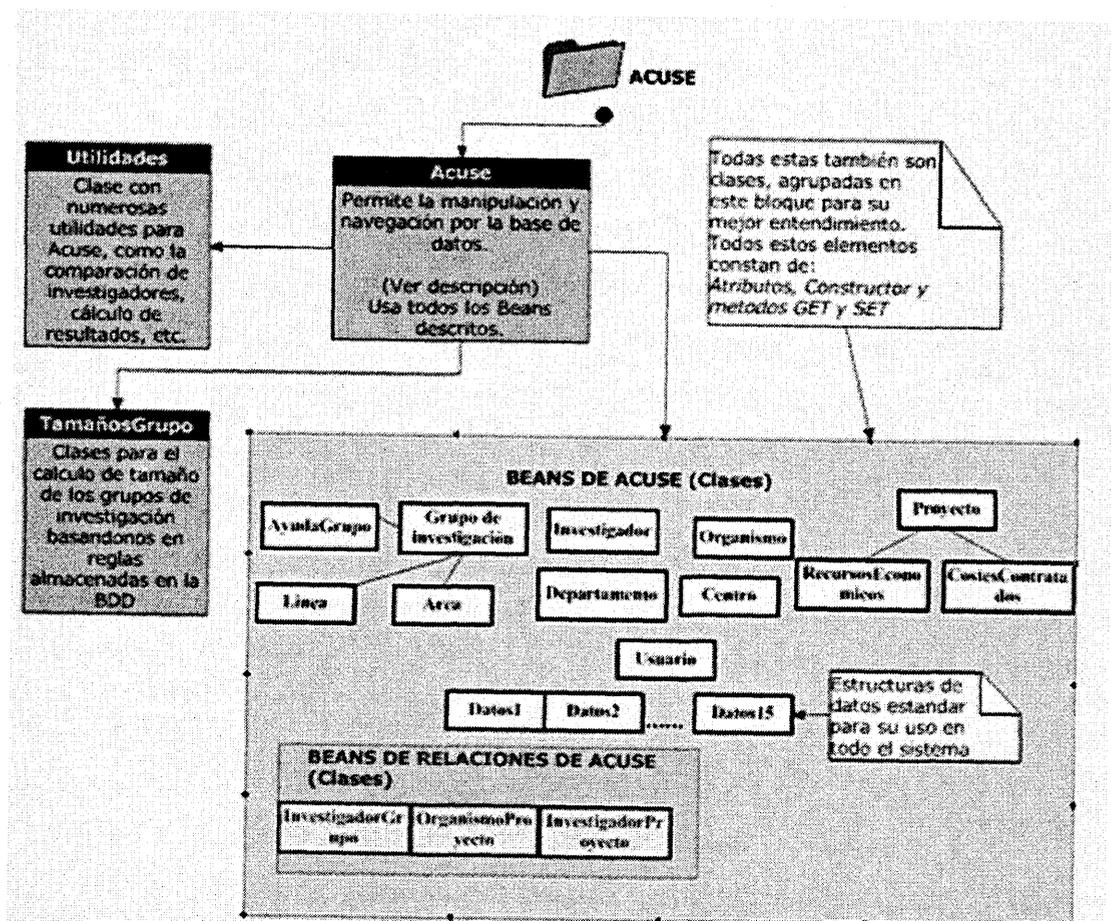
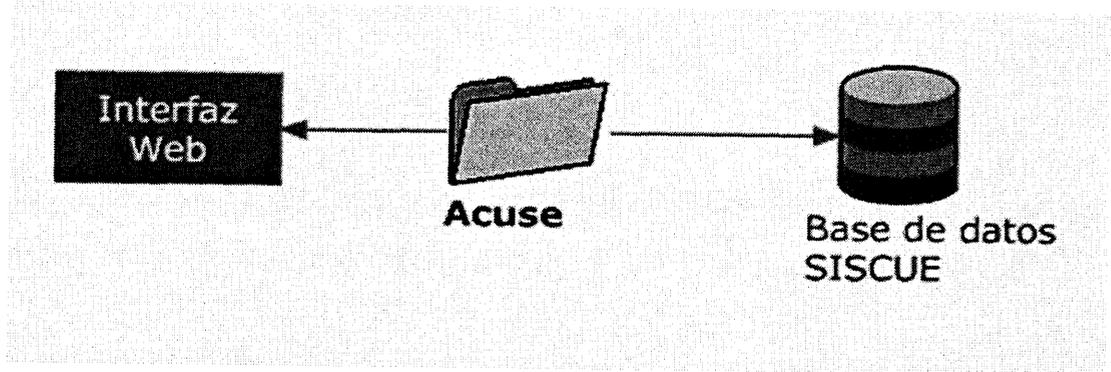


FIGURA 4-8: DIAGRAMA DE PAQUETES Y CLASES DE ACUSE

Tiene numerosas estructuras con las que funciona, todas del tipo de objeto estándar Java (atributos y métodos get set). Además cuenta con una clase principal que usando todos los demás maneja por completo el paquete.

Descripción de clases: Separamos por un lado (al igual que en el diagrama de clases) las clases de manejo y los beans (Estructuras)

### Clases más importantes del paquete ACUSE<sup>1</sup>:

**Acuse.java:** Para ver la descripción en detalle de todos sus métodos, se puede consultar el javadoc. Grosso modo las operaciones que realiza son:

- **Lectura** (navegación) de todos los elementos de la BDD (Proyectos, organismos, líneas, áreas, departamentos, etc.) pasándole un identificador. Nos devuelve la estructura pedida.

#### **Ejemplo de métodos:**

- *Proyecto* *getProyecto(String id\_proyecto)*: Devuelve un proyecto con todos sus datos de la base de datos, pasándole el id
- *ArrayList* *getOrganismos(String tipo,int id\_tipo,String pais,String orden)*: Devuelve un array de organismos pasándole qué tipo queremos, (tanto genérico como el subtipo) y en qué orden lo queremos.
- **Agregado** de elementos a la bdd: pasándole un elemento, nos lo inserta de una manera lógica (encargándose de inicializar todos los parámetros y relaciones, y validando la información antes de ser agregada) en la bdd.

#### **Ejemplo de métodos:**

- *int* *AgregarInvestigador(Investigador inv)*: agrega el objeto investigador a la base de datos.
- *String* *AgregarGrupoOficial(GrupoInvestigacion grupo)*: Agrega un grupo de investigación a la base de datos.

---

<sup>1</sup> Para ver con detalle los métodos y estructura de los paquetes, consultar los javadocs que se pueden encontrar en la carpeta Docs de la aplicación Siscue.

- **Borrado** de elementos en la bdd: Borra un elemento seleccionado, teniendo en cuenta las restricciones y medidas de seguridad de datos implementadas en la aplicación<sup>2</sup>

**Ejemplo de métodos:**

- *boolean BorrarOrganismo(int id\_organismo)* Borra el organismo del que le pasamos el id, un borrado no es trivial, conlleva el análisis y borrado de todas sus relaciones y de ajustar todo para que no quede "basura" en la base de datos.

- **Modificación** de elementos en la bdd, tanto de sus parámetros como la modificación, agregado o borrado de relaciones.

**Ejemplo de métodos:**

- *boolean ModificarOrganismo(Organismo org):* Modifica los parámetros deseados extraídos del objeto que recibe como parámetro.
- *int ModificarCategoria(String id\_grupo,int id\_inv,int cod\_categ,String quefecha,String tipo,boolean desactivado):* Modifica desde un administrador de grupos de investigación con permisos de personal, la categoría de un miembro y los datos relacionados con la relación investigador-grupo de investigación.

**TamañosGrupos.java:** Clase usada para la definición de reglas de cálculo de tamaños de grupos, que luego son importadas por otros paquetes como el Plan de calidad.

**Utilidades.java:** Diferentes utilidades usadas para facilitar algunos servicios a la clase Acuse, como por ejemplo la búsqueda de parecidos o la concatenación de cadenas, etc..

---

<sup>2</sup> Aparte de las medidas que toma la bdd para la seguridad y validación de datos, la aplicación java impone un filtro con todas sus medidas de validación de datos, comprobación de relaciones, permisos de acceso, etc.

#### 4.3.1.2.2 Módulo Consultas

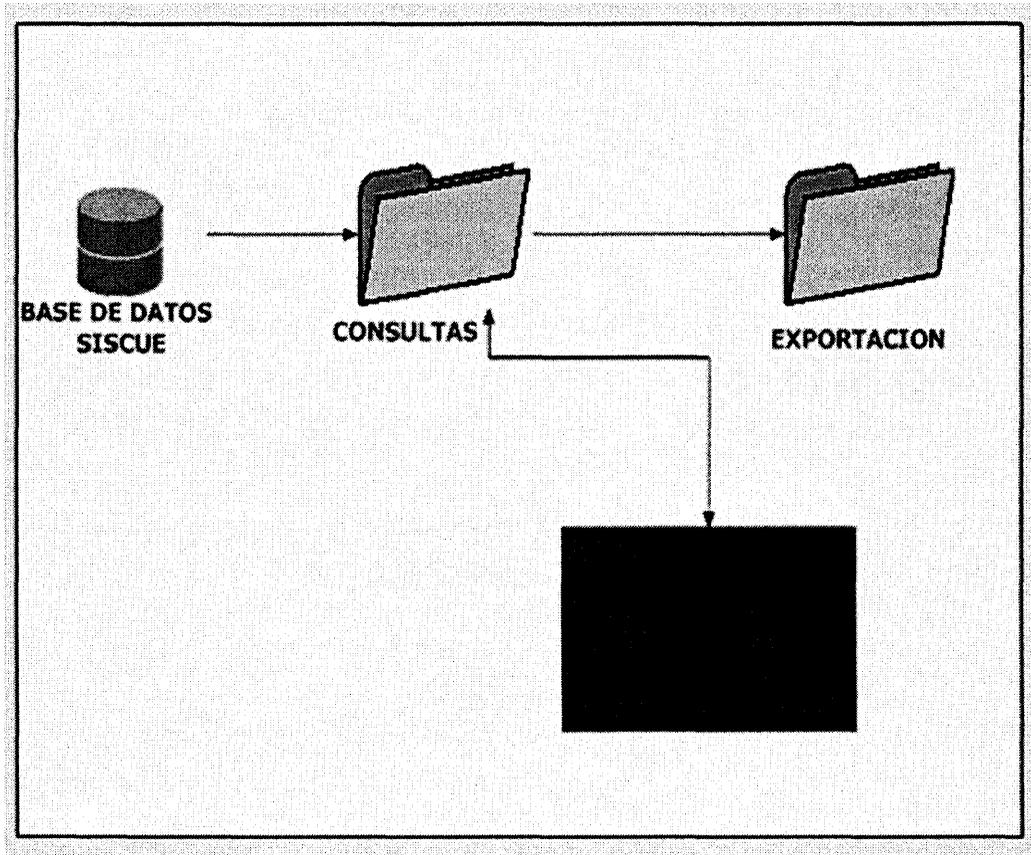


FIGURA 4-9. DIAGRAMA DE PAQUETES DE CONSULTAS

Es el encargado de la generación y normalización de cualquier tipo de consulta relativamente sencilla sobre proyectos, creando unas consultas “tipo” con una estructura común para el cruzado de todo tipo de elementos.

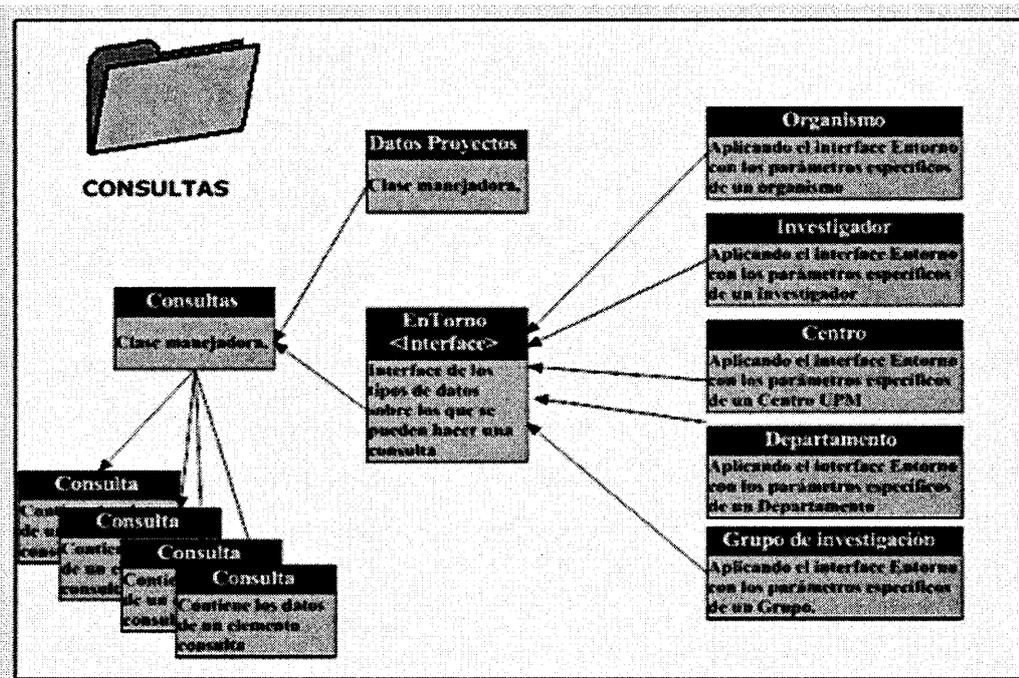


FIGURA 4-810: DIAGRAMA DE CLASES DE CONSULTAS

Como se puede observar, las clases principales de las que se compone este paquete son:

- **Consultas:** Clase principal, recoge un elemento en torno al cual se va a hacer la consulta con todos los filtros de la misma, por ejemplo, una consulta entorno a países llevará un objeto "Organismos" que hereda de Entorno, configurando el orden, el tipo de organismo, etc.
- **DatosProyectos:** Configura toda la información relativa a los proyectos que queremos consultar (marco, año, presupuesto, etc.)
- **Consulta:** Estructura de la consulta, el resultado de usar este paquete es un array de elementos "Consulta" con todos los datos recogidos de la consulta configurada
- **EnTorno:** Es una clase abstracta que define las reglas para los elementos que vamos a configurar para elaborar la consulta.

Y el resto son clases que heredan de la estructura tipo "EnTorno", especificando las reglas para cada elemento (si la consulta va en torno a Organismos, o Investigadores, etc.) Las clases son:

- **Investigador**
- **Centro**
- **Departamento**
- **Grupo de investigación**
- **Organismo**

Solo mostramos le código de la clase manejadora "Consultas.java" ya que es la más importante, en ConsultasV2 se definen más clases que amplían este concepto y permiten ver la forma de realizar este tipo de consultas de una manera mucho mas completa.

-----Inicio Consultas.java-----

```
/**
 * Consultas para Siscue.
 *
 * @version 1.2
 * @since JDK1.5
 * @author Miguel Angel Lucas del Pozo
 */

public class Consultas {

    public void setEnTorno(EnTorno n){
        this.entorno=n;
    }
    public void setDatosProyectos(DatosProyectos d){
        this.datosP=d;
    }
    public EnTorno getEnTorno(){
        return(this.entorno);
    }
    public DatosProyectos getDatosProyectos(){
        return(this.datosP);
    }
    //CONSULTAS!!!

    public ArrayList getConsulta() throws Exception{

        Statement stmt=conn.createStatement();
        ArrayList consultas=new ArrayList();

        //Transformación de datos.
        String paraproys="";
        String presup="";
        String ambitoGeo="";
        String paraMarco="";
        String paraAnio="";

        //tipo de proyectos, años y marco
        String tipoProyectos=this.datosP.getTipo();
        int marco=this.datosP.getMarco();
        String anio=this.datosP.getAnios();

        if(!anio.equals("2003-2006")){
            paraAnio="AND(p.anio='"+anio+"'");
        }

        if(!tipoProyectos.equals("Todos")){
            ambitoGeo="AND(m.ambitogeografico_marcocolab='"+tipoProyectos+"'");
            if(marco!=0){
                paraMarco="AND(m.id_marcocolab='"+marco+"'");
            }
        }
        //Presupuesto
        String presupuesto=this.datosP.getPresupuesto();
```

```

if(presupuesto.equals("Todos")){
presup="";
}
if(presupuesto.equals("<6.000")){
presup="AND(re.presupuesto_concedido_upm<6000)";
}
if(presupuesto.equals("6.000-60.000")){
presup="AND((re.presupuesto_concedido_upm>6000)AND(re.presupuesto_conc
edido_upm<60000))";
}
if(presupuesto.equals("60.000-600.000")){
presup="AND((re.presupuesto_concedido_upm>60000)AND(re.presupuesto_con
cedido_upm<600000))";
}
if(presupuesto.equals(">600.000")){
presup="AND(re.presupuesto_concedido_upm>600000)";
}

String
paraConsulta=this.entorno.getParaConsulta(paraproys,ambitoGeo,paraMarc
o,paraAnio,presup);

String paraOrden=this.entorno.getOrden();
System.out.println("CONSULTA: "+paraConsulta+paraOrden);
ResultSet resultado=stmt.executeQuery(paraConsulta+paraOrden);

while(resultado.next()){
String dato=resultado.getString("dato");
double cantidad=resultado.getFloat("cantidad");
int id_utilizable=resultado.getInt("id_utilizable");
String_Codigo_utilizable=resultado.getString("Codigo_utilizable");
Consulta eyy=new
Consulta(dato,cantidad,id_utilizable,Codigo_utilizable);
consultas.add(eyy);
}
return(consultas);

}
} //FIN

```

-----Fin Consultas.java-----

Vamos a configurar una consulta de proyectos en torno a organismos, para ello configuramos primero el objeto Organismos, y luego todas las opciones de consulta, por último le pedimos al sistema que nos la genere, y nos devuelve un ArrayList (java.util) con objetos "Consulta":

-----Generación manual de consulta en torno a organismos -----

Este ejemplo está sacado del interface web del SISCUE, donde se ha eliminado la gestión de errores para su mejor entendimiento.

```
Consultas cons=new Consultas();
cons.conectar();
//Configuramos los datos de los proyectos
//para la consulta
DatosProyectos datosP=new DatosProyectos();
datosP.setTipo("Europeo");
datosP.setMarco(marco);
datosP.setPresupuesto("Todos");
datosP.setAnios("2003-2006");
cons.setDatosProyectos(datosP);
//Configuramos el objeto en torno al cual
//vamos a realizar la consulta
Organismos entorno2=new Organismos();
entorno2.setPais("Todos");
entorno2.settipoOrganismo("Todos");
entorno2.setTipo("numproys");
entorno2.setOrden("cantidad DESC");
//Se introducen en la aplicación todos los datos configurados
cons.setEnTorno(entorno2);
System.out.println("Prueba2");
ArrayList consultas=new ArrayList();
try{
cons.conectar();
consultas=cons.getConsulta();
}catch(Exception e){
//Gestion de error
}

entorno2.setTipo("sumapre");
```

```
cons.setEnTorno(entorno2);
ArrayList consultas2=new ArrayList();
try{
consultas2=cons.getConsulta();
}catch(Exception e){
//Gestion de error
}
```

-----Fin generación de consulta-----

#### 4.3.1.2.3 **Módulo ConsultasV2**

Este módulo sirve para generar consultas complejas especificando cada tipo de elemento sobre el que se realiza las consultas, generando tablas configurables con todos los datos requeridos. Tiene implementado dos elementos: Proyectos y Grupos de Investigación, aunque está preparada para ampliarse con más. La enorme diferencia entre este módulo y el de Consultas, es que el módulo V2 (Versión 2) aporta estas ampliaciones:

- Recibe un array de todos los campos que se quiere buscar y mostrar en la tabla
- Se configura todos los datos sobre los que quiere que se busque (del elemento, de sus relaciones, etc...)
- Genera una tabla bidimensional con todo lo seleccionado compatible con el módulo de exportación a Excel, creando links a los elementos que referencia

Si se quiere ver su uso, se muestra algún ejemplo en la parte del interface web, donde se utiliza este módulo.

La estructura de este módulo es la misma que la de Consultas, salvo que el elemento de consulta es una Array y no un solo objeto.

La clase manejadora es **ConsultasV2.java**, que conecta con la base de datos, y que una vez configurados todos los objetos para la consulta, devuelve el resultado final:

----- Inicio ConsultasV2.java-----

```
/**
 * Consultas para Siscue.
 * @version 1.2
 * @since JDK1.5
 * @author Miguel Angel Lucas del Pozo
 */

public class ConsultasV2 {

    static String bd="acuse";
    static String login="ma.lucas";
    static String password="jean1982";
    static String url="jdbc:mysql://localhost:3306/"+bd;
    Connection conn;
    String sentencia="";

    DatosProyectosV2 datosP;
    EntornoV2 entorno;

    public ConsultasV2(){

    }

    /**Conecta con la base de datos Acuse de LocalHost (Modificable en el
    codigo fuente)
    **/
    public String conectar() {
    try {
    System.out.println("E");
    Driver d =
    (Driver)Class.forName("com.mysql.jdbc.Driver").newInstance();
    conn=DriverManager.getConnection(url, login, password);
    }catch (Exception a){
    System.out.println("Ha habido una excepcion"+a.getMessage());
    }
    return("bien");

    }

}

} //de conectar()
```

```
public void setEntorno(EntornoV2 n){
    this.entorno=n;
}
public void setDatosProyectos (DatosProyectosV2 d){
    this.datosP=d;
}
public EntornoV2 getEntorno(){
    return(this.entorno);
}
public DatosProyectosV2 getDatosProyectosV2(){
    return(this.datosP);
}
//CONSULTAS!!!

public ConsultaV2 getConsulta() throws Exception{

    Statement stmt=conn.createStatement();
    //Transformación de datos.
    String paraproys="";
    String presup="";
    String ambitoGeo="";
    String paraMarco="";
    String paraAnio="";
    String paraCadena="";

    String cadena=this.datosP.getCadena();
    if(!cadena.equals("")){
        paraCadena="AND(p.titulo_proyecto like '%" +cadena+"%'";
    }

    //tipo de proyectos, años y marco
    String tipoProyectos=this.datosP.getTipo();
    int marco=this.datosP.getMarco();
    String anio=this.datosP.getAnios();

    if(!anio.equals("2003-2006")){
        paraAnio="AND(p.anio='"+anio+"')";
    }
    if(anio.equals("Todos")){
        paraAnio="";
    }
}
```

```

        if(!tipoProyectos.equals("Todos")){

            ambitoGeo="AND(m.ambitogeografico_marcocolab='"+tipoProyectos+"
        )";

            if(marco!=0){
                paraMarco="AND(m.id_marcocolab="+marco+" ";
            }

        }

        //Presupuesto
        String presupuesto=this.datosP.getPresupuesto();
        if(presupuesto.equals("Todos")){
            presup="";
        }
        if(presupuesto.equals("<6.000")){
            presup="AND(re.presupuesto_concedido_upm<6000)";
        }
        if(presupuesto.equals("6.000-60.000")){
            presup="AND((re.presupuesto_concedido_upm>6000)AND(re.presupuesto_conc
            edido_upm<60000))";
        }
        if(presupuesto.equals("60.000-600.000")){
            presup="AND((re.presupuesto_concedido_upm>60000)AND(re.presupuesto_con
            cedido_upm<600000))";
        }
        if(presupuesto.equals(">600.000")){
            presup="AND(re.presupuesto_concedido_upm>600000)";
        }

        String
        paraConsulta=this.entorno.getParaConsulta(paraproys, ambitoGeo, paraMarc
        o, paraAnio, presup, paraCadena);

        String paraOrden=this.entorno.getOrden();
        ResultSet resultado=stmt.executeQuery(paraConsulta+paraOrden);
        ArrayList total=new ArrayList();

        ArrayList valores=this.entorno.getRecogerSelecciones();
        total.add(valores);
        while(resultado.next()){
            valores=this.entorno.getRecogerSelecciones();

```

```
ArrayList valores2=new ArrayList();
for(int i=0;i<valores.size();i++){
    String valor=(String)valores.get(i);
    String valor2=resultado.getString(valor);

    valores2.add(valor2);

}
total.add(valores2);

}
ConsultaV2 eyy=new ConsultaV2();
eyy.setDatos(total);
return(eyy);

}

} //FIN
```

-----Fin ConsultasV2.java-----

**ConsultaV2.java** no es más que un tipo de estructura de array bidimensional, que contendrá el resultado de toda la consulta creada

-----Inicio ConsultaV2.java-----

```
/**
 * Consultas V2 para Siscue.
 * IMPORTANTE: el parámetro es un ArrayList bidimensional (arrayList
 con un arrayList
 * para cada columna, el elemento 0 son los nombres de los valores, y
 los siguientes su valores por el orden
 * propuesto
 * @version 1.2
 * @since JDK1.5
 * @author Miguel Angel Lucas del Pozo
 */
```

```
public class ConsultaV2 {

    private ArrayList datos;

    public ArrayList getDatos() {
        return datos;
    }

    public void setDatos(ArrayList datos) {
        this.datos = datos;
    }
}
```

-----Fin ConsultaV2.java-----

**ProyectosV2.java** es un objeto que pasándole los parámetros sobre los proyectos que queremos configurar, genera todo lo necesario para poder posteriormente realizar la consulta.

-----Inicio ProyectosV2.java-----

```
public class ProyectosV2 extends EnTornoV2{

    private String paraOrgas;
    private String condicionOrgas;
    private String paraInvs;
    private int nacionalidad;
    private String selecciones;
    private String setOrgas;
    private ArrayList recogerSelecciones;
    private String Centro;
    private String Departamento;
    private String paraIP;
    public ProyectosV2(){

        super.Orden="";
        System.out.println("CREAMOS EL PROYECTOS2");
        this.recogerSelecciones=new ArrayList();
    }

    //Metodos heredados
    public void setOrden(String s){
        if(s.equals("cantidad ASC")){
            super.Orden=s;
        }
        if(s.equals("cantidad DESC")){
            super.Orden=s;
        }
        if(s.equals("Alf")){
            super.Orden="p.titulo_proyecto";
        }
    }
    public void setTipo(String s){
```

```

        super.tipo=s;
    }
    public String getOrden(){
        if(!super.Orden.equals("")){
            return(" order by "+super.Orden);
        }else{
            return("");
        }
    }
    public String getTipo(){
        return(super.tipo);
    }

    public String getParaConsulta(String paraproys,String ambitoGeo,String
    paraMarco,String paraAnio,String presup,String paraCadena){
        //Construcción de la consulta
        String paraCentros="";
        if(!this.Centro.equals("Todos")){
            paraCentros="AND(dep.Centro='"+this.Centro+"')";
        }
        String paraDepartamentos="";
        if(!this.Departamento.equals("Todos")){
            paraDepartamentos="AND(dep.codigo='"+this.Departamento+"')";
        }

        String esip="";
        if(this.paraIP.equals("si")){

            this.paraInvs="inner join invpro ip on p.id_proyecto=ip.id_proyecto" +
            " inner join investigador i on ip.id_investigador=i.id_investigador";
            esip="AND(ip.es_responsable=1)";
        }else{
            this.paraInvs="";
        }

        if(super.tipo.equals("numproys")){
            return("SELECT " +this.selecciones+" " +
            "FROM proyecto as p "+this.paraOrgas+this.paraInvs+" " +
            "inner join tipodeproyecto as tp on
            p.id_tipo_proyecto=tp.id_tipo_proyecto " +
            "inner join recursos_economicos as re on p.id_rree=re.id_rree " +

```

```
"inner join marcocolaboracion m on p.id_marco=m.id_marcocolab " +
"inner join programa as prog on p.id_programa=prog.id_programa " +
"inner join instrumento as inst on p.id_inst=inst.id_instrumento " +
"inner join area a on p.id_area=a.id_area " +
"inner join departamentos dep on p.codigo_departamento=dep.codigo " +
"inner join centros cen on dep.centro=cen.codigo_centro " +
"Where(true)"+paraCadena+paraproys+ambitoGeo+paraMarco+paraAnio+presup
+paraCentros+paraDepartamentos+esip+this.condicionOrgas);
}
//DONE
    if(super.tipo.equals("sumapre")){
        return("");
    }
    return("");
}

    public String getParaOrgas() {
        return paraOrgas;
    }

    //tipo (Todos si son todos), id_tipo (-1 si son todos),pais
    (Todos si son todos,Nacionales, Extranjeros), paisespecifico
    public void setParaOrgas(String e,String tipo,int id_tipo,String
    pais,String paisespecifico) {

        this.setOrgas=e;
        this.condicionOrgas="";
        if(e.equals("si")){
            this.paraOrgas=" left join organismoproyecto op on " +
            "p.id_proyecto=op.id_proyecto left join " +
            "organismo o on op.id_organismo=o.id_organismo " +
            "left join tipoorganismo torg on
            o.id_tipo_organismo=torg.id_tipo_organismo " +
            "left join pais as pais on pais.id_pais=o.id_pais " +
            "left join comunidades com on com.id_comunidad=o.id_comunidad ";
            if(id_tipo!=-1){
                this.condicionOrgas=this.condicionOrgas+"AND(o.id_tipo_organismo="+id_
                tipo+")";
            }
            if(!tipo.equals("Todos")){
                this.condicionOrgas=this.condicionOrgas+"AND(torg.Tipo_organismo_tipo_
                organismo2='"+tipo+"')";
            }
        }
    }
}
```

```

}
if(pais.equals("Nacionales")){
this.condicionOrgas=this.condicionOrgas+"AND(o.id_pais=15)";
}
if(pais.equals("Extranjeros")){
    if(paisespecifico.equals("Todos Extranjero")){
        this.condicionOrgas=this.condicionOrgas+"AND(o.id_pais!=15)";
    }else{
        this.condicionOrgas=this.condicionOrgas+"AND(pais.pais='"+paises
pecifico+"'");
    }
}
}else{
this.paraOrgas = "";
}
}

public int getNacionalidad() {
    return nacionalidad;
}

public void setNacionalidad(int nacionalidad) {
    this.nacionalidad = nacionalidad;
}

public String getSelecciones() {
    return selecciones;
}

public void setSelecciones(String selecciones) {
    this.selecciones = selecciones;
}

public void crearSelecciones(ArrayList sele){
ArrayList seleccionesFinal=new ArrayList();
this.recogerSelecciones=new ArrayList();

for(int i=0;i<sele.size();i++){
String s=(String)sele.get(i);

if(s.equals("titulo")){

seleccionesFinal.add("p.titulo_proyecto as Titulo");

```

```
this.recogerSelecciones.add("Titulo");
}
if(s.equals("tipo")){
seleccionesFinal.add("tp.Tipo_proyecto as Tipo");
recogerSelecciones.add("Tipo");
}
if(s.equals("codigo")){
seleccionesFinal.add("p.codigo_proyecto asCodigoOTT");
recogerSelecciones.add("CodigoOTT");
}
if(s.equals("centro")){
seleccionesFinal.add("cen.centro as Centro");
this.recogerSelecciones.add("Centro");
}
if(s.equals("ip")){
seleccionesFinal.add("concat(i.nombre_investigadorip,'
',i.apellidos_investigadorip) as InvestigadorResponsable");
this.recogerSelecciones.add("InvestigadorResponsable");
}
if(s.equals("departamento")){
seleccionesFinal.add("dep.departamento as Departamento");
recogerSelecciones.add("Departamento");
}
if(s.equals("ambgeo")){
seleccionesFinal.add("m.ambitogeografico_marcocolab as
AmbitoGeografico");
recogerSelecciones.add("AmbitoGeografico");
}
if(s.equals("presu")){
seleccionesFinal.add("re.presupuesto_concedido_upm as
PresupuestoConcedidoUPM");
recogerSelecciones.add("PresupuestoConcedidoUPM");
}
if(s.equals("marco")){
seleccionesFinal.add("m.convocatoria_marcocolab as
MarcoColaboracion");
recogerSelecciones.add("MarcoColaboracion");
}
if(s.equals("programa")){
seleccionesFinal.add("prog.programa as Programa");
recogerSelecciones.add("Programa");
}
```

```

}
if(s.equals("anio")){
seleccionesFinal.add("p.anio as Anio");
recogerSelecciones.add("Anio");
}
if(s.equals("instrumento")){
seleccionesFinal.add("inst.instrumento_nombre as Instrumento");
recogerSelecciones.add("Instrumento");
}
if(s.equals("observaciones")){
seleccionesFinal.add("p.observaciones_proyecto as Observaciones");
recogerSelecciones.add("Observaciones");
}
if(s.equals("area")){
seleccionesFinal.add("a.area as Area");
recogerSelecciones.add("Area");
}
//Si se han seleccionado ORGANISMOS se rellenan
if(this.setOrgas.equals("si")){
if(s.equals("nombreOrg")){
seleccionesFinal.add("o.nombre_organismo as NombreOrganismo");
recogerSelecciones.add("NombreOrganismo");
}
if(s.equals("CifOrg")){
seleccionesFinal.add("o.cif_organismo as CIF_Organismo");
recogerSelecciones.add("CIF_Organismo");
}if(s.equals("tipoOrg1")){
seleccionesFinal.add("torg.Tipo_organismo_tipo_organismo2 as
TipoOrganismo");
recogerSelecciones.add("TipoOrganismo");
}if(s.equals("tipoOrg2")){
seleccionesFinal.add("torg.Especif_tipo_organismo2 as
TipoEspecificoOrganismo");
recogerSelecciones.add("TipoEspecificoOrganismo");
}if(s.equals("PaisOrg")){
seleccionesFinal.add("pais.pais as PaisOrganismo");
recogerSelecciones.add("PaisOrganismo");
}if(s.equals("contactoOrg")){
seleccionesFinal.add("o.Direccion_calle_organismo as
Calle_Organismo");
recogerSelecciones.add("Calle_Organismo");
}

```

```
seleccionesFinal.add("o.Direccion_codigo_postal_organismo as
CP_Organismo");
recogerSelecciones.add("CP_Organismo");
seleccionesFinal.add("o.Direccion_ciudad_organismo as
Ciudad_Organismo");
recogerSelecciones.add("Ciudad_Organismo");
seleccionesFinal.add("o.Direccion_provincia_organismo as
Provincia_Organismo");
recogerSelecciones.add("Provincia_Organismo");
seleccionesFinal.add("com.comunidad as Comunidad_Organismo");
recogerSelecciones.add("Comunidad_Organismo");
seleccionesFinal.add("o.telefono_organismo as Telefono_Organismo");
recogerSelecciones.add("Telefono_Organismo");
seleccionesFinal.add("o.web_organismo as Web_Organismo");
recogerSelecciones.add("Web_Organismo");
seleccionesFinal.add("o.Fax_organismo as Fax_Organismo");
recogerSelecciones.add("Fax_Organismo");
    }
    }
}
this.crearSeleccionesFinal(seleccionesFinal);
}

    public void crearSeleccionesFinal(ArrayList selecciones){
        if(selecciones.size()>0){
            this.selecciones=(String)selecciones.get(0);
            for(int i=1;i<selecciones.size();i++){
                String s=(String)selecciones.get(i);
                this.selecciones=this.selecciones+" "+s;
            }
        }else{
            this.selecciones="";
        }
    }

    public ArrayList getRecogerSelecciones() {
        return recogerSelecciones;
    }

    public void setRecogerSelecciones(ArrayList recogerSelecciones)
    {
        this.recogerSelecciones = recogerSelecciones;
    }
}
```

```
}

public String getCentro() {
    return Centro;
}

public void setCentro(String centro) {
    Centro = centro;
}

public String getCondicionOrgas() {
    return condicionOrgas;
}

public void setCondicionOrgas(String condicionOrgas) {
    this.condicionOrgas = condicionOrgas;
}

public String getDepartamento() {
    return Departamento;
}

public void setDepartamento(String departamento) {
    Departamento = departamento;
}

public String getParaInvs() {
    return paraInvs;
}

public void setParaInvs(String paraInvs) {
    this.paraInvs = paraInvs;
}

public String getSetOrgas() {
    return setOrgas;
}

public void setSetOrgas(String setOrgas) {
    this.setOrgas = setOrgas;
}
}
```

```
public void setParaOrgas(String paraOrgas) {  
    this.paraOrgas = paraOrgas;  
}  
  
public String getParaIP() {  
    return paraIP;  
}  
  
public void setParaIP(String paraIP) {  
    this.paraIP = paraIP;  
}  
  
}
```

-----fin ProyectosV2.java-----

Teniendo en cuenta las clases anteriores y sus parámetros, un posible uso sería:

#### //CREACION DE LA CONSULTA

```
ConsultasV2 cons=new ConsultasV2();
cons.conectar();

DatosProyectosV2 datosP=new DatosProyectosV2();
datosP.setCadena(INPUTcadena);
datosP.setTipo(tipoProyecto);
datosP.setMarco(marcocolaboracion);
datosP.setPresupuesto(presupuesto);
datosP.setAnios(anio);
cons.setDatosProyectos(datosP);

ProyectosV2 entorno2=new ProyectosV2 ();
entorno2.setTipo("numproys");
entorno2.setOrden("Alf");
entorno2.setCentro(centroSelect);
entorno2.setDepartamento(departamentoSelect);
entorno2.setParaIP(queremosip);
entorno2.setParaOrgas(verOrgas,tipo,id_tipo,pais,paisespecifico);
entorno2.crearSelecciones(selecciones);

cons.setEnTorno(entorno2);
ConsultaV2 consultas=cons.getConsulta();

ArrayList total=(ArrayList)consultas.getDatos();
```

Donde el array total sería un array bidimensional con todos los resultados deseados.

### 4.3.1.2.4 Módulo GruposC

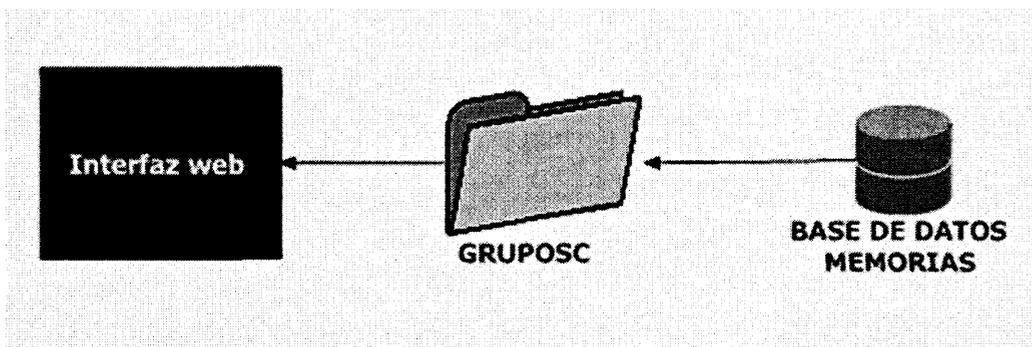


FIGURA 4-11: DIAGRAMA DE PAQUETES DE GRUPOSC

Al igual que la clase Acuse.java para la base de datos acuse, gruposC es un paquete destinado a las mismas labores sobre la base de datos de memorias de investigación, reuniendo sobre todo, la navegación sobre los datos existentes.

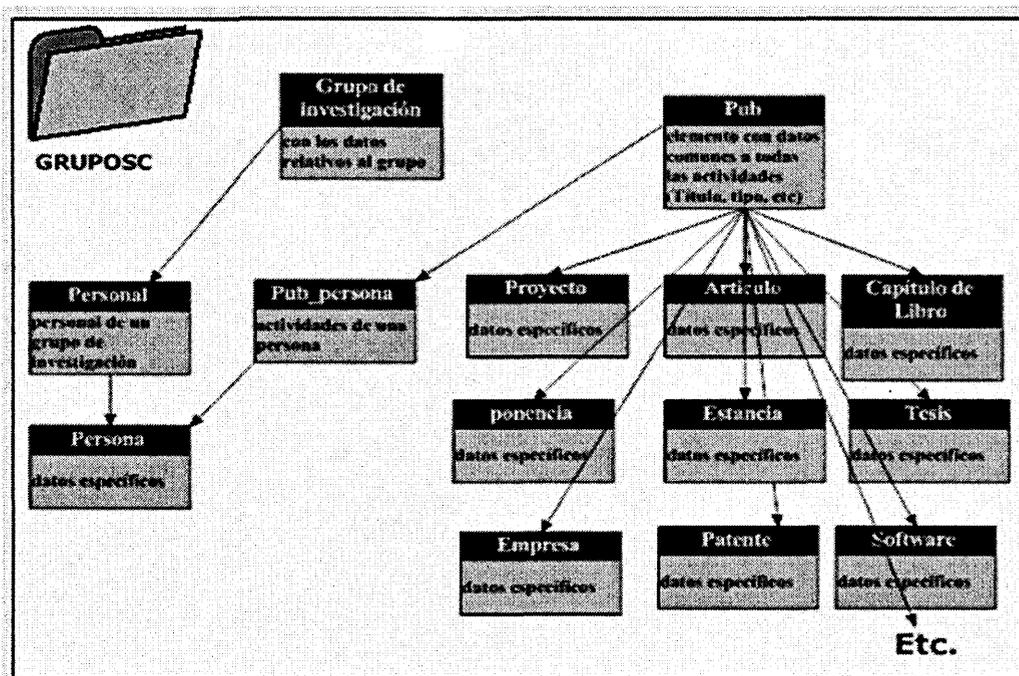


FIGURA 4-12: DIAGRAMA DE CLASES DE GRUPOSC

GruposC tiene el mismo tipo de estructura que Acuse.java, donde existe una clase central para el uso de toda la información, numerosos beans de estructuras que recopilan todos los tipos de datos de la base, y alguna clase para utilidades o agregados.

### 4.3.1.2.5 Módulo Exportacion

Encargado de exportar tablas sacadas de consultas a Excel, o gráficas a pdf.

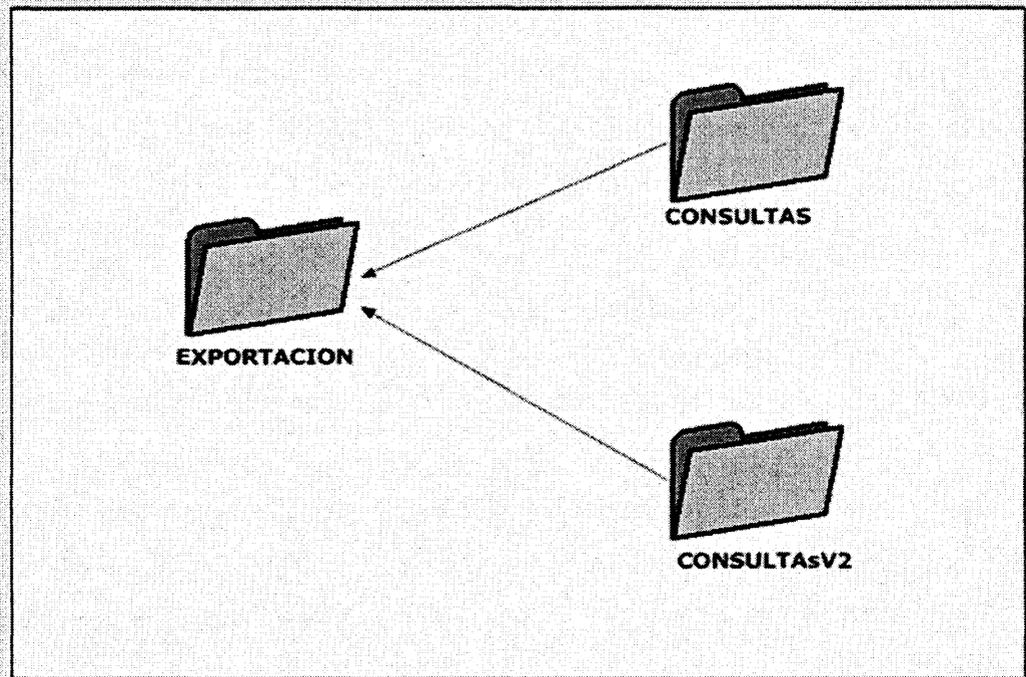


FIGURA 4-13: DIAGRAMA DE PAQUETES DE CONSULTASV2

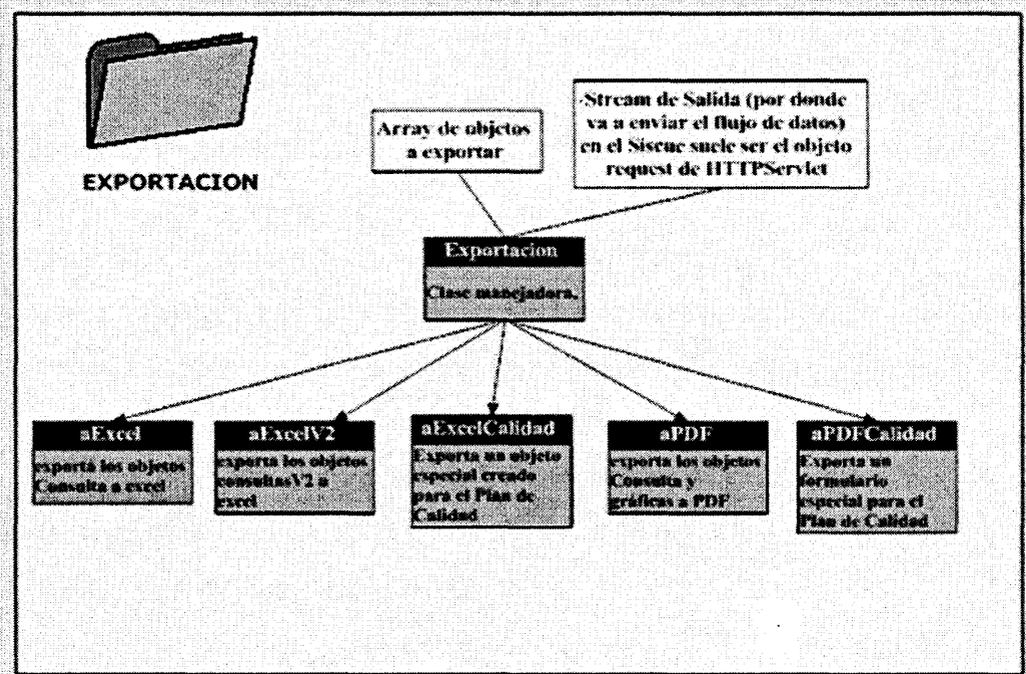


FIGURA 4-14: DIAGRAMA DE CLASES DE CONSULTASV2

#### **4.3.1.2.6 Módulo PlanCalidad**

Realiza todo el plan de calidad, cooperando con grupos, el módulo PlanCalidad realiza un gran número de operaciones estadísticas basándose en el “*Plan de Calidad de la Investigación (PGCI)*” aplicado a los grupos de investigación, aprobado en consejo de gobierno en 31 de Mayo de 2007. Para ello tiene una clase principal “PlanCalidad.java” que usando el módulo gruposC para la petición base de datos, genera este plan de manera configurable (parametrizado).

El plan de calidad tiene la siguiente estructura de indicadores:

##### **1. Generación de recursos económicos Peso = 6/27**

- 1.1. Cantidad ingresada por proyectos y convenios en convocatorias públicas competitivas (Peso = 12/27)
- 1.2. Cantidad ingresada por proyectos y convenios de financiación pública no competitiva (Peso = 6/27)
- 1.3. Cantidad ingresada por contratos y convenios de financiación privada (Peso = 6/27)
- 1.4. Otros ingresos (Peso = 3/27)

##### **2. Formación de investigadores y movilidad Peso = 5/27**

- 2.1. Tesis doctorales dirigidas por algún PDI de la unidad (Peso = 4/14)
- 2.2. Becarios y contratados para la realización del Doctorado (Peso = 5/14)
- 2.3. Contratos de personal investigador (Peso = 3/14)
- 2.4. Estancias y sabáticos recogidos en programas oficiales (Peso = 2/14)

Plan General de la Calidad de la Investigación Mayo 2007

Aplicación a Grupos de Investigación

17/25

##### **3. Difusión de resultados de investigación Peso = 7/27**

- 3.1. Libros o capítulos en libros (excluyendo los de texto) (Peso = 2/34)
- 3.2. Publicaciones en revistas internacionales incluidas en

- el SCI con índice de impacto alto (mitad superior) (Peso = 10/34)
- 3.3. Publicaciones en revistas internacionales incluidas en el SCI con menor índice de impacto (mitad inferior) (Peso = 6/34)
- 3.4. Publicaciones en otras revistas con revisores (Peso = 2/34)
- 3.5. Ponencias en congresos internacionales con revisores (Peso = 6/34)
- 3.6. Ponencias en congresos nacionales con revisores (Peso = 1/34)
- 3.7. Informes para las AAPP o sus organismos dependientes (Peso = 1/34)
- 3.8. Conferencias invitadas en congresos internacionales (Peso = 6/34)

#### **4. Explotación de resultados y transferencia de tecnología Peso = 3/27**

- 4.1. Patentes con extensión internacional en explotación (Peso = 12/42)
- 4.2. Patentes con extensión internacional sin explotación (Peso = 6/42)
- 4.3. Patentes de ámbito nacional en explotación (Peso = 7/42))
- 4.4. Patentes de ámbito nacional sin explotación (Peso = 2/42)
- 4.5. Registros de software en explotación (Peso = 7/42)
- 4.6. Empresas creadas (Peso = 8/42)

#### **5. Reconocimiento de méritos Peso = 6/27**

- 5.1. Número de sexenios (sexenios totales x sexenios relativos)

Teniendo en cuenta todos estos grupos de indicadores, el módulo PlanCalidad va recogiendo todas las consultas en torno a esos indicadores, de la información sacada de la base de datos de memorias de investigación, ponderando y realizando los cálculos pertinentes para la elaboración de dicho plan.

Como resultado final aparte de configurar todo este plan, tenemos una estructura compuesta de todos los datos necesarios para la realización de gráficas, calculo de valores absolutos, etc... (para ver los resultados, ver la parte de plan de calidad en el interfaz web).

La cabecera de este método del cálculo del Plan de Calidad de la Investigación es:

```
public ArrayList getConsultaPlanCalidad(String anio,String ver,String Consolidados,boolean Cerrados,int id_primer,int id_segundo,int origen_tam_investigadores, int dividendo);
```

**Donde los parámetros son:**

**String anio:** Año o años de los que se quiere sacar el Plan de calidad.

**String ver:** qué queremos ver, solo grupos de indicadores, solo resultados finales, todo, etc.

**String Consolidados:** Tipo de grupos, consolidados, en proceso, etc.

**boolean Cerrados:** realizar el plan solo sobre grupos cerrados.

**int id\_primer:** id del grupo de indicadores sobre el que queremos realizar el plan (0 todos).

**int id\_segundo:** dentro de un grupo de indicadores si queremos todos este id deberá ir a 0, sino, se especifica.

**int origen\_tam\_investigadores:** Calcular el número de investigadores de la base de datos acuse o de la de memorias de investigación.

**int dividendo:** Para reducir la unidad monetaria sobre la que se calcula, si se pone algo como 15.000, la unidad de dinero será 15000€, no 1€.

Como se puede observar, algunos de estos campos van como cadena de caracteres sin forzar la entrada, esto es porque se recogen de la base de datos, así, con solo cambiar en la base de datos un tipo de grupo, de indicadores, o de lo que se necesite, el método sigue funcionando perfectamente.

El **ArrayList** que nos devuelve es un array de elementos ConsultaPC, descrito en los javadocs, donde cada objeto ConsultaPC tiene la siguiente estructura:

-----INICIO ConsultaPC.java-----

//CLASE ConsultaPC

```
public class ConsultaPC {  
  
    String nombre_grupo;  
    String id_grupo;  
    ArrayList grupoIndicadores;  
    double totalGrupo;  
    int tam_grupo;  
    int tam_grupoMemorias;  
    boolean oficial;  
  
    public ConsultaPC(){  
  
        this.grupoIndicadores=new ArrayList();  
    }  
  
    public String getId_grupo() {  
        return id_grupo;  
    }  
  
    public void setId_grupo(String id_grupo) {  
        this.id_grupo = id_grupo;  
    }  
  
    public ArrayList getGrupoindicadores() {  
        return grupoIndicadores;  
    }  
  
    public void setGrupoindicadores(ArrayList indicadores) {  
        grupoIndicadores = indicadores;  
    }  
}
```

```
public String getNombre_grupo() {  
    return nombre_grupo;  
}
```

```
public void setNombre_grupo(String nombre_grupo) {  
    this.nombre_grupo = nombre_grupo;  
}
```

```
public double getTotalGrupo() {  
    return totalGrupo;  
}
```

```
public void setTotalGrupo(double totalGrupo) {  
    this.totalGrupo = totalGrupo;  
}
```

```
public int getTam_grupo() {  
    return tam_grupo;  
}
```

```
public void setTam_grupo(int tam_grupo) {  
    this.tam_grupo = tam_grupo;  
}
```

```
public int getTam_grupoMemorias() {  
    return tam_grupoMemorias;  
}
```

```
public void setTam_grupoMemorias(int tam_grupoMemorias) {  
    this.tam_grupoMemorias = tam_grupoMemorias;  
}
```

```

public boolean isOficial() {
    return oficial;
}

public void setOficial(boolean oficial) {
    this.oficial = oficial;
}

public void inicializar(){

    PlanCalidad pc=new PlanCalidad();
    pc.conectar();

    ArrayList arrayGruposIndicadores=new ArrayList();

    //1
    GrupoIndicador grupoIndicador=new GrupoIndicador();
    grupoIndicador.setIld_primer(1);
    ArrayList indicadoresgrupo=new ArrayList();
    Indicador indi=pc.getIndicador(11);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
        indi=pc.getIndicador(12);
        indi.setCantidad(0);
        indicadoresgrupo.add(indi);
        indi=pc.getIndicador(13);
        indi.setCantidad(0);
        indicadoresgrupo.add(indi);
        indi=pc.getIndicador(14);
        indi.setCantidad(0);
        indicadoresgrupo.add(indi);

    grupoIndicador.setIndicadores(indicadoresgrupo);
    arrayGruposIndicadores.add(grupoIndicador);
    //2
    grupoIndicador=new GrupoIndicador();
    grupoIndicador.setIld_primer(2);
    indicadoresgrupo=new ArrayList();
    indi=pc.getIndicador(21);

```

```
indi.setCantidad(0);
indicadoresgrupo.add(indi);
    indi=pc.getIndicador(22);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(23);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(24);
    indi.setCantidad(0);
indicadoresgrupo.add(indi);

grupoIndicador.setIndicadores(indicadoresgrupo);
arrayGruposIndicadores.add(grupoIndicador);
//3
grupoIndicador=new GrupoIndicador();
grupoIndicador.setId_primer(3);
indicadoresgrupo=new ArrayList();
    indi=pc.getIndicador(31);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(32);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(33);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(34);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(35);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(36);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(37);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(38);
```

```
        indi.setCantidad(0);
        indicadoresgrupo.add(indi);

    grupoIndicador.setIndicadores(indicadoresgrupo);
    arrayGruposIndicadores.add(grupoIndicador);
```

```
//4
```

```
grupoIndicador=new GrupoIndicador();
grupoIndicador.setId_primer(4);
indicadoresgrupo=new ArrayList();
    indi=pc.getIndicador(41);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(42);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(43);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(44);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(45);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
    indi=pc.getIndicador(46);
    indi.setCantidad(0);
    indicadoresgrupo.add(indi);
```

```
grupoIndicador.setIndicadores(indicadoresgrupo);
arrayGruposIndicadores.add(grupoIndicador);
```

```
//5
```

```
//4
```

```
grupoIndicador=new GrupoIndicador();
grupoIndicador.setId_primer(5);
indicadoresgrupo=new ArrayList();
```

```
        indi=pc.getIndicador(51);
        indi.setCantidad(0);
        indicadoresgrupo.add(indi);

        grupoIndicador.setIndicadores(indicadoresgrupo);
        arrayGruposIndicadores.add(grupoIndicador);

        this.grupoIndicadores=arrayGruposIndicadores;
    }

    public double getCantidadEspecifica(int id_segundo){

        ArrayList GruposdeIndicadores=this.getGrupoindicadores();

        for(int k=0;k<GruposdeIndicadores.size();k++){

            GrupoIndicador g_ind_g=(GrupoIndicador)GruposdeIndicadores.get(k);

            ArrayList indicadores=g_ind_g.getIndicadores();

            for(int m=0;m<indicadores.size();m++){
                Indicador ind_grupo=(Indicador)indicadores.get(m);
                //DATos de un indicador
                int id_segundo2=ind_grupo.getId_indicador();

                if(id_segundo==id_segundo2){
                    double cantidad=ind_grupo.getCantidad();
                    return(cantidad);
                }
            }

        }

    }

    return(0);
}
```

```

public void setCantidadEspecifica(int id_segundo ,double cantidad){
try{
ArrayList GruposdeIndicadores=this.grupoIndicadores;

for(int k=0;k<GruposdeIndicadores.size();k++){

    GrupoIndicador g_ind_g=(GrupoIndicador)GruposdeIndicadores.get(k);

    ArrayList indicadores=g_ind_g.getIndicadores();

        for(int m=0;m<indicadores.size();m++){
            Indicador ind_grupo=(Indicador)indicadores.get(m);
            //DATos de un indicador
            int id_segundo2=ind_grupo.getId_indicador();

            if(id_segundo==id_segundo2){

                ind_grupo.setCantidad(cantidad);
                indicadores.set(m,ind_grupo);
            }

            GruposdeIndicadores.set(k,g_ind_g);
        }
        this.grupoIndicadores=GruposdeIndicadores;
    }
}catch(Exception e){
    System.out.println("Error en getCantidadEspecifica de ConsultaPC:"+e.getMessage());
}
}
}
}

```

-----FIN ConsultaPC.java-----

Dentro de esta clase hay un ArrayList de los grupos de indicadores, que son objetos GrupoIndicador. Java:

-----Inicio GrupoIndicador.java-----

```
package gruposC;
import java.util.*;

public class GrupoIndicador {

    ArrayList indicadores;
    int id_primer;
    double valor;

    public GrupoIndicador(){
        this.valor=0;
    }

    public int getId_primer() {
        return id_primer;
    }

    public void setId_primer(int id_primer) {
        this.id_primer = id_primer;
    }

    public ArrayList getIndicadores() {
        return indicadores;
    }

    public void setIndicadores(ArrayList indicadores) {
        this.indicadores = indicadores;
    }

    public double getValor() {
```

```

        return valor;
    }

    public void setValor(double valor) {
        this.valor = valor;
    }
}
-----Fin GrupoIndicador.java-----

```

Y, por ultimo, se puede ver que este grupo tiene un ArrayList de indicadores, objetos Indicador.java:

```

-----Inicio Indicador.java-----

package gruposC;

public class Indicador {

    private int id_indicador;
    private int id_primero;
    private String descripcionPrimero;
    private String descripcionSegundo;
    private int peso_indicador;
    private int peso_grupo;
    private double cantidad;
    private double cantidad_original;
    private int peso_total_indicadores;

    public Indicador(){

    }

    public String getDescripcionPrimero() {
        return descripcionPrimero;
    }

    public void setDescripcionPrimero(String descripcionPrimero) {
        this.descripcionPrimero = descripcionPrimero;
    }
}

```

```
}  
public String getDescripcionSegundo() {  
    return descripcionSegundo;  
}  
public void setDescripcionSegundo(String descripcionSegundo) {  
    this.descripcionSegundo = descripcionSegundo;  
}  
public int getId_indicador() {  
    return id_indicador;  
}  
public void setId_indicador(int id_indicador) {  
    this.id_indicador = id_indicador;  
}  
public int getId_primer() {  
    return id_primer;  
}  
public void setId_primer(int id_primer) {  
    this.id_primer = id_primer;  
}  
public int getPeso_grupo() {  
    return peso_grupo;  
}  
public void setPeso_grupo(int peso_grupo) {  
    this.peso_grupo = peso_grupo;  
}  
public int getPeso_indicador() {  
    return peso_indicador;  
}  
public void setPeso_indicador(int peso_indicador) {  
    this.peso_indicador = peso_indicador;  
}  
  
public double getCantidad() {  
    return cantidad;  
}  
  
public void setCantidad(double cantidad) {  
    this.cantidad = cantidad;  
}
```

```

    public double getCantidad_original() {
        return cantidad_original;
    }

    public void setCantidad_original(double cantidad_original) {
        this.cantidad_original = cantidad_original;
    }

    public int getPeso_total_indicadores() {
        return peso_total_indicadores;
    }

    public void setPeso_total_indicadores(int peso_total_indicadores) {
        this.peso_total_indicadores = peso_total_indicadores;
    }

}

-----Fin Indicador.java-----

```

Si se estudian estas clases, se puede ver que a cada grupo de investigación se le asigna un grupo de indicadores, pero que para asignarlo, primero se deberá hacer el cálculo sobre todos, ya que, por ejemplo, para tener todos los resultados sobre un grupo primero hay que conocer el de todos para poder sacar un valor ponderado.

Una vez entendido el sistema de datos usado para recoger el plan de calidad, una forma de pedir el plan de calidad y exponer los datos puede ser:

-----INICIO EJEMPLO USO PLAN DE CALIDAD -----

```

//Primero recogemos todos los datos para configurar el plan de calidad, en este
ejemplo se //pasan directamente a mano en la llamada al método:

```

```

PlanCalidad pc=new PlanCalidad();

```

```

ArrayList Consulta =

```

```

pc.getConsultaPlanCalidad("2006","Todo","Consolidado",true,1,0,"Oficial",1500
0);

```

```
//Recogida de los datos del Array

for(int i=((pagina-1)*cantidadamostrar);i<limite;i++){

    ConsultaPC consult=(ConsultaPC)Consulta.get(i);
    String nombre_grupo=consult.getNombre_grupo();
    String id_grupo=consult.getId_grupo();
    int tam_grupo_memorias=consult.getTam_grupoMemorias();
    int tam_grupo=consult.getTam_grupo();
    int tam_grupo_usado=0;
    if(origen_tam_investigadores==1)
        tam_grupo_usado=tam_grupo;
    if(origen_tam_investigadores==2)
        tam_grupo_usado=tam_grupo_memorias;

    ArrayList GruposdelIndicadores=consult.getGrupoidicadores();

    double total_grupo=consult.getTotalGrupo();
    double total_grupo_peso=total_grupo/fullcalidad;
    total_grupo_peso=pc.redondear(total_grupo_peso,9);
    double porcentaje=0;
    String paratotalporcentaje="";

    porcentaje=total_grupo_peso*100;
    porcentaje=pc.redondear(porcentaje,5);
    sumacomprobadora1=sumacomprobadora1+porcentaje;
    int num=i+1;

for(int k=0;k<GruposdelIndicadores.size();k++){

    GrupoIndicador g_ind_g=(GrupoIndicador)GruposdelIndicadores.get(k);

    int id_Clasificador=g_ind_g.getId_primer();
    if((id_primer==0)||id_Clasificador==id_primer){
        ArrayList indicadores=g_ind_g.getIndicadores();
    Indicador paraver= pc.getDatosIndicadorPrimerNivel(id_Clasificador);
    int peso_grupo=paraver.getPeso_grupo();
    String descripcionprimero=paraver.getDescripcionPrimer();
    double valortotalenprimero=g_ind_g.getValor();
```

```

if(modo==2){
    valortotalenprimero=valortotalenprimero*100;
    valortotalenprimero=pc.redondear(valortotalenprimero,5);

}
for(int m=0;m<indicadores.size();m++){
    Indicador ind_grupo=(Indicador)indicadores.get(m);
    //DATos de un indicador

String descripcionsegundo=ind_grupo.getDescripcionSegundo();
double cantidad=ind_grupo.getCantidad();

int id_primero2=ind_grupo.getId_primero();
int id_segundo2=ind_grupo.getId_indicador();

if((id_segundo==0)||(id_segundo2==id_segundo)){

int peso_indicador=ind_grupo.getPeso_indicador();
int cantidadporgrupo=ind_grupo.getPeso_total_indicadores();

double valor_indicador= pc.calcularpeso(cantidad,peso_indicador,cantidadporgrupo);
    valor_indicador=pc.redondear(valor_indicador,5);
    double absoluto2=cantidad*tam_grupo_usado;
    absoluto2=pc.redondear(absoluto2,2);
    String absoluto="";
    if(id_Clasificador==1){
        NumberFormat nf = NumberFormat.getInstance();
        absoluto2=absoluto2*dividendo;
        absoluto=nf.format(absoluto2);
        absoluto=absoluto+" €";
    }else{
        absoluto=new Double(absoluto2).toString();
    }

}

} //For indicadores
} //for grupos de indicadores
} //for grupos de investigación

```

-----FIN EJEMPLO USO PLAN DE CALIDAD-----

#### 4.3.1.2.7 **Módulo Servlet: GeneracionGraficasSiscue**

Servlet que recibiendo objetos Consulta genera graficas configurables. Para el uso de este servicio, el servlet recibe por el objeto "session" de jsp un array de objetos "Consulta" del módulo visto anteriormente, y el tipo de gráfica que se desea.

También puede recibir otros tipos de parámetros (Colores, etc.) opcionales, para una configuración más detallada.

Para el desarrollo de la generación de gráficas se barajaron diferentes librerías, seleccionando al final JFreeChart por su relativa sencillez y su documentación, descrito al inicio de este capítulo.

Esta librería usa otras típicas de java como java.awt para diseño, en en fichero .java podemos ver los imports usados.

Todo el servlet se recoge en un solo fichero GraficasSiscue.java, detallado a continuación:

```
-----INICIO GraficasSiscue.java-----  
/**  
 * GraficasServlet.java  
 * Created on 7 de noviembre de 2006, 10:41  
 * @author Miguel Angel Lucas del Pozo  
 * @version 2.1  
 **/  
//Lista de imports:  
import java.lang.*;  
import java.io.*;  
import java.net.*;  
import java.util.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
import org.jfree.data.*;
```

```

import org.jfree.data.xy.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PiePlot;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;
import org.jfree.data.category.*;
import org.jfree.chart.axis.*;
import org.jfree.chart.renderer.category.*;
import java.sql.*;
import Consultas.*;
import java.awt.*;

```

```

public class GraficasSiscue extends HttpServlet

```

```

{

```

```

    Connection conn=null;

```

```

/**

```

```

 * Método para crear la gráfica, pasándole los datos

```

```

 * de configuración y las consultas

```

```

**/

```

```

public JFreeChart crearChart(String tipo,ArrayList consultas,int limite,String titulo)

```

```

{

```

```

    if (limite>consultas.size())

```

```

    {

```

```

        limite=consultas.size();

```

```

    }

```

```

    //QUESO

```

```

    if(tipo.equals("Queso")){

```

```

        if(consultas.size(>)limite)

```

```

        {

```

```
        limite=limite-1;
    }

    DefaultPieDataset dataset = new DefaultPieDataset();

    try{
        double totaldato=0;
        for(int i=0;i<consultas.size();i++){

            Consulta consult=(Consulta)consultas.get(i);
            double dato2int=consult.getcantidad();
            totaldato=totaldato+dato2int;

        }

        for(int i=0;i<limite;i++){

            Consulta consult=(Consulta)consultas.get(i);

            double dato2int=consult.getcantidad();
            String dato1=consult.getdato();
            double porcentaje2=(dato2int*100)/totaldato;
            int porcentaje=new Double(porcentaje2).intValue();
            dato1=dato1+": "+porcentaje+"%";
            dataset.setValue(dato1, dato2int);

        }

        //Necesitamos el campo OTROS
        double cantidadotros=0;
        if(consultas.size()>limite){

            for(int i=limite;i<consultas.size();i++){
                //Sumamos todos en otros
                Consulta consult=(Consulta)consultas.get(i);
                double dato2int=consult.getcantidad();
                cantidadotros=cantidadotros+dato2int;

            }//for

            double porcentaje2=(cantidadotros*100)/totaldato;
```

```

        int porcentaje=new Double(porcentaje2).intValue();
        String otrosName="Otros= "+cantidadotros+"\n"+porcentaje+"%";
        dataset.setValue(otrosName, cantidadotros);

    }
}catch(Exception e){System.out.println("EXCEPCION"+e.getMessage());}

JFreeChart chart = ChartFactory.createPieChart(
    titulo, // chart title
    dataset, // los datos antes recogidos
    false, // Si queremos legenda
    false,
    false
);

    chart.setNotify(true);
    chart.setBorderVisible(false);
    return chart;
}

//BARRAS

if(tipo.equals("Barras")){

    DefaultCategoryDataset dataset = new DefaultCategoryDataset();

    try{
        for(int i=0;i<limite;i++){

            Consulta consult=(Consulta)consultas.get(i);
            String dato1=consult.getdato()+" ";
            double dato2int=consult.getcantidad();
            dataset.addValue( dato2int, "", dato1 );

        }
    }catch(Exception e){
        //No tratamos esta excepcion
        System.out.println("EXCEPCION"+e.getMessage());
    }
}

```

```
JFreeChart chart =ChartFactory.createBarChart3D(  
    titulo,  
    "",  
    "",  
    dataset,  
    PlotOrientation.HORIZONTAL,  
    false,  
    true,  
    true);  
    return chart;  
}  
  
if(tipo.equals("Barras2")){  
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();  
    System.out.println("Barras2");  
    try{  
        for(int i=0;i<limite;i++){  
            Consulta consult=(Consulta)consultas.get(i);  
            String dato1=consult.getdato()+" ";  
            double dato2int=consult.getcantidad();  
            dataset.addValue( dato2int, "", dato1 );  
        }  
    }catch(Exception e){  
        System.out.println("EXCEPCION"+e.getMessage());  
    }  
}
```

```
JFreeChart chart =ChartFactory.createBarChart(  
    titulo,  
    "",  
    "",  
    dataset,  
    PlotOrientation.VERTICAL,  
    false,  
    false,  
    false);
```

```

final CategoryPlot plot = chart.getCategoryPlot();
    plot.setBackgroundPaint(Color.lightGray);
    plot.setDomainGridlinePaint(Color.white);
    plot.setRangeGridlinePaint(Color.white);

// set the range axis to display integers only...
final NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());

// disable bar outlines...
final BarRenderer renderer = (BarRenderer) plot.getRenderer();
renderer.setDrawBarOutline(false);

// set up gradient paints for series...
final GradientPaint gp0 = new GradientPaint(
    0.0f, 0.0f, Color.blue,
    0.0f, 0.0f, Color.lightGray
);
final GradientPaint gp1 = new GradientPaint(
    0.0f, 0.0f, Color.green,
    0.0f, 0.0f, Color.lightGray
);
final GradientPaint gp2 = new GradientPaint(
    0.0f, 0.0f, Color.red,
    0.0f, 0.0f, Color.lightGray
);
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(1, gp1);
renderer.setSeriesPaint(2, gp2);

final CategoryAxis domainAxis = plot.getDomainAxis();

domainAxis.setCategoryLabelPositions(CategoryLabelPositions.createUpRotationLabelPosition
s(Math.PI / 6.0));

    return chart;
}
if(tipo.equals("Barras3")){

```

```
DefaultCategoryDataset dataset = new DefaultCategoryDataset();  
try{  
    for(int i=0;i<limite;i++){  
  
        Consulta consult=(Consulta)consultas.get(i);  
        String dato1=consult.getdato()+" ";  
        double dato2int=consult.getcantidad();  
        dataset.addValue( dato2int, "", dato1 );  
  
    }  
}catch(Exception e){  
    System.out.println("EXCEPCION"+e.getMessage());  
}
```

```
        JFreeChart chart =ChartFactory.createBarChart3D(  
            titulo,  
            "",  
            "",  
            dataset,  
            PlotOrientation.HORIZONTAL,  
            false,  
            true,  
            true);  
    return chart;  
}
```

```
if(tipo.equals("tipo3")){  
  
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();  
    System.out.println("tipo3");  
  
    try{  
        for(int i=0;i<limite;i++){  
  
            Consulta consult=(Consulta)consultas.get(i);  
            String dato1=consult.getdato();  
            String dato2=consult.getCodigo();  
            double dato2int=consult.getcantidad();
```

```

        dataset.addValue( dato2int, dato2, dato1 );

    }

}catch(Exception e){
    System.out.println("EXCEPCION"+e.getMessage());
}

        JFreeChart chart =ChartFactory.createBarChart(
            titulo,
            "",
            "",
            dataset,
            PlotOrientation.VERTICAL,
            true,
            true,
            true);

    final CategoryPlot plot = chart.getCategoryPlot();
    final CategoryAxis domainAxis = plot.getDomainAxis();

    domainAxis.setCategoryLabelPositions(CategoryLabelPositions.createUpRotationLabelPosition
s(Math.PI / 6.0));
        return chart;
    }

//LINEAL 2
if(tipo.equals("Lineas2")){

DefaultCategoryDataset dataset = new DefaultCategoryDataset();

    try{
        for(int i=0;i<limite;i++){

            Consulta consult=(Consulta)consultas.get(i);
            String dato1=consult.getdato()+" ";

```

```
        double dato2int=consult.getcantidad();
        //AHORA NO METEMOS LOS DATOS SALE SOLO LA GRAFICA
        dataset.addValue( dato2int, "", new Integer(i+1).toString() );
    }
}catch(Exception e){
    System.out.println("EXCEPCION"+e.getMessage());
}
```

```
        JFreeChart chart =ChartFactory.createLineChart(
            titulo,
            "",
            "",
            dataset,
            PlotOrientation.VERTICAL,
            false,
            true,
            true);

        return chart;
    }//fin lineas2
if(tipo.equals("Area3")){

    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    try{
        for(int i=0;i<limite;i++){

            Consulta consult=(Consulta)consultas.get(i);
            String dato1=consult.getdato()+" ";
            double dato2int=consult.getcantidad();
            //AHORA NO METEMOS LOS DATOS SALE SOLO LA GRAFICA
            dataset.addValue( dato2int, "", new Integer(i+1).toString() );

        }
    }catch(Exception e){System.out.println("EXCEPCION"+e.getMessage());}
```

```
        JFreeChart chart =ChartFactory.createAreaChart(
            titulo,
            "",
            "",
```

```

        dataset,
        PlotOrientation.VERTICAL,
        false,
        true,
        true);

    return chart;
} //fin lineas2

//FIN DEL METODO
return(null);}

//Más métodos de la clase:

int getParamEntero(HttpServletRequest request, String pNombre, int pDefecto)
{
    String param = request.getParameter(pNombre);

    if (param == null || param.compareTo("") == 0)
    {
        return pDefecto;
    }

    return Integer.parseInt(param);
}

/** Processes requests for both HTTP GET and POST methods.
 * @param request servlet request
 * @param response servlet response
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("image/jpeg");
    //Tratamos como queremos sacar el grafico (modificable para la salida)
    OutputStream salida = response.getOutputStream();
    //Conseguimos compartir la session
    HttpSession session = request.getSession(true);

```

```
ArrayList cons=new ArrayList();
    try{
        cons=(ArrayList)session.getAttribute("Consultas");
    }catch(Exception e){
        System.out.println("No hay consulta ");
    }
String tipoGrafica=request.getParameter("tipoGraficas");
String titulo=request.getParameter("titulo");
int limite=10;
String limiteGraficas=request.getParameter("CantidadGrafica");
if(limiteGraficas!=null){
    limite=new Integer(limiteGraficas).intValue();
}

JFreeChart grafica = crearChart(tipoGrafica,cons,limite,titulo);
    grafica.setBackgroundImageAlignment(0);
    grafica.setBackgroundPaint(Color.WHITE);

int ancho = getParamEntero(request,"ancho",560);
int alto = getParamEntero(request,"alto",340);
ChartUtilities.writeChartAsJPEG(salida,grafica,ancho,alto);

salida.close();
}

/** Handles the HTTP GET method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("DoGeT");
    processRequest(request, response);
}
}

-----FIN GraficasSiscue.java-----
```

### 4.3.2 Interfaz Web

Como interfaz al usuario, se ha elegido uno Web, debido a las siguientes ventajas:

- El navegador actual soporta códigos empotrados de la calidad de java (JSP) con librerías existentes de compatibilidad con MySQL.
- El diseño para conseguir un interfaz amigable gracias a HTML y objetos de diseño es relativamente sencillo.
- Amplia gama de librerías free ya construidas para todo tipo de utilidades.
- Interfaz más conocido actualmente por el usuario
- Accesible desde cualquier pc con conexión sin necesidad de software nuevo
- Simplicidad de la centralización de la aplicación, http y todo lo relacionado con web ya nos crea un sistema distribuido.

### Proceso de diseño y desarrollo del interfaz web:

2. **Esquema de navegación:** Qué hojas vamos a necesitar y las características de cada una (De forma genérica), dividiendo 2 grandes bloques:
  - a. **Navegación:** por todos los tipos de datos, y con todos los tipos de consultas contempladas.
  - b. **Administración:** óptima por parte del elemento administrador, de tal forma que de una manera amigable pueda modificar, borrar y agregar todos los tipos de datos
3. **Diseño gráfico base:** Que se usará en todo el interfaz.
4. **Especificación (y programación) de los elementos comunes** (Barra de navegación superior e izquierda, y elementos que van a ser usados por la mayoría de nuestro interfaz).
5. **Especificación (y programación) del resto de hojas,** para cada requerimiento de los servicios que va a ofrecer.

### 4.3.2.1 Diagrama de navegación y hojas creadas: Un viaje por la web

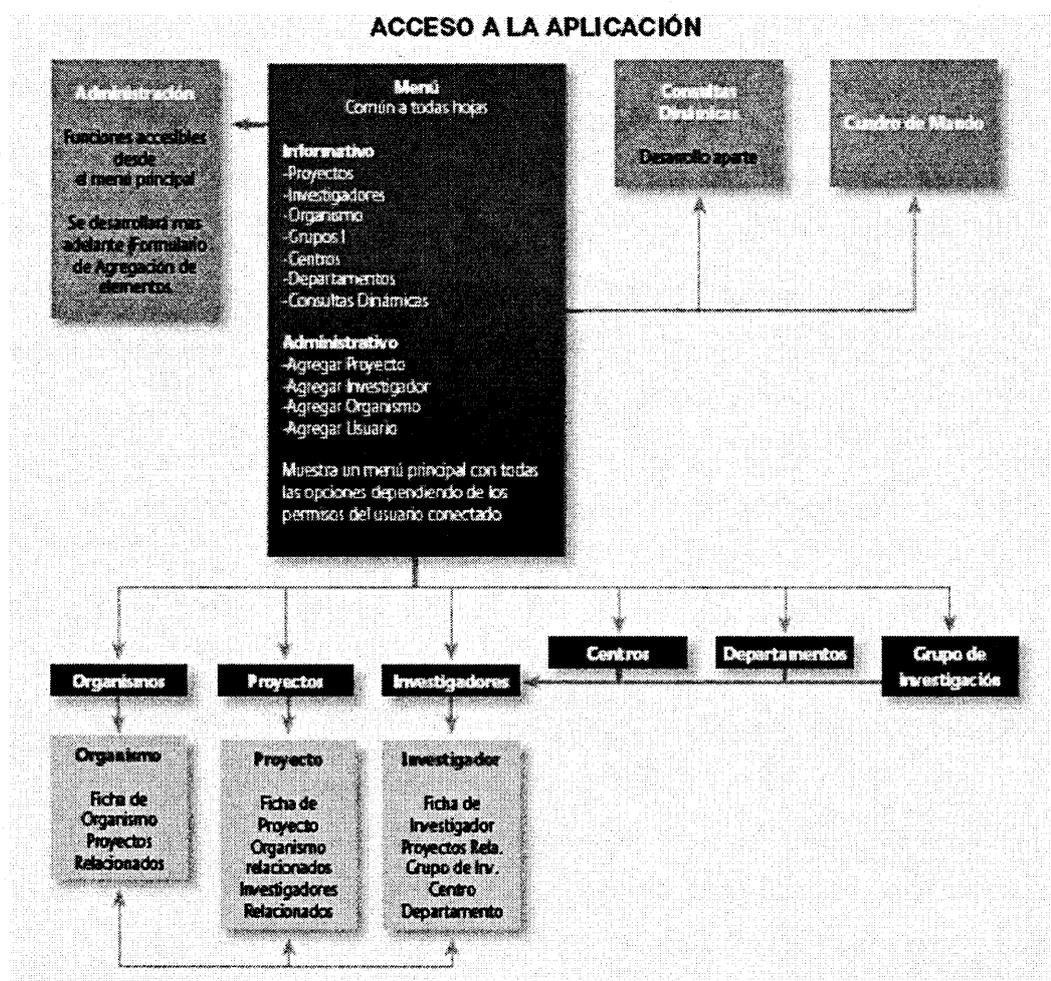


FIGURA 4-15: DIAGRAMA DE NAVEGACIÓN DEL INTERFAZ WEB

Visto este diagrama de navegación nos podemos hacer una pequeña idea de cómo movernos por el sistema, partiendo de un menú principal donde se nos conserva en todo momento los elementos de navegación más importantes en una barra superior y lateral.

En la barra lateral tenemos buscadores de los elementos más importantes de la base, proyectos, grupos de investigación, organismos, investigadores y actividades de investigación. También se añaden links a los cuadros de mando y a los elementos de administración más importantes. Todo esto pasa por un filtro de permisos, de tal forma que si un usuario no tiene permiso para navegar

por los proyectos no le aparece dicho buscador, o para los usuarios normales desaparecen todos los elementos de administración.

**BUSCADORES**

**Proyectos:**  
 » 

**Organismos:**  
 »

**Invs. en grupos de inv.:**  
 »

**Investigadores:**  
 »

**Grupos Inv :**  
 » 

**Actividades Grupos:**  
 »

para una búsqueda avanzada de elementos pinchar en el más.

---

**Miguel Angel Lucas**  
[Perfil](#)

**Cuadros de Mando:**

**Europeos:**

**Grupos de inv:**

---

[Inicio](#)  
[Desconectar](#)

En la imagen de la barra lateral también se puede observar un “más” a la derecha de los elementos proyectos y grupos de investigación (se prevé tenerlo para todos los elementos de los que consta la base de datos). Este “más” nos abre un panel de búsqueda avanzada de dicho elemento, de tal forma que podemos configurar de arriba abajo todo tipo de consultas sobre ello seleccionando lo que queremos que nos muestre. Una vez hecho esto podemos llevarnos los resultado a Excel para un trabajo en local.

Esta barra es un archivo .jsp (BarraLateral.jsp) importada por todas las hojas del sistema en esa posición, de tal forma que para cualquier añadido o modificación de la barra basta con hacerlo en un solo archivo, actualizándose para toda la plataforma (este elemento común está en constante crecimiento debido al aumento del sistema entero).

El link “Perfil” nos permite ver un perfil entero del usuario que ha accedido dejando modificar datos como nombre y contraseña, y mostrando toda la configuración de permisos del usuario.

La hoja BarraLateral.jsp se muestra a continuación como ejemplo de jsp implementado en el sistema, las demás hojas solo se mostrará parte ya que el volumen de líneas de código es excesivo.

-----Inicio BarraLateral.jsp-----

```
<%@page contentType="text/html" import="java.util.*,java.lang.*,Acuse.*" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Documento sin título</title>
<SCRIPT language=JAVASCRIPT>
function abrir(){
window.open("CuadroDeMandoE.jsp?Marco=8&tipo=Europeo","cuadro","toolbar=no,location=no,
directories=no,status=no,menubar=no,scrollbar=no,resizable=no,width=750,height=645");}

function abrir2(){
window.open("PlanCalidad.jsp?Cerrado=Si&Consolidado=Si");
}
</SCRIPT>
<link rel="STYLESHEET" href="styles.css" type="text/css" />
<style type="text/css">
</style>
</head>
<body>
<%
//Recuperar el tipo de usuario
Acuse c=new Acuse();
String conn=c.conectar();
Integer idUsu2=(Integer)session.getAttribute("idUsu");
int idUsu=idUsu2.intValue();
Usuario usu=c.getUsuario(idUsu);
String name=usu.getNombre_completo();
Permisos per=c.getPermisos(idUsu);

%>
<div class="Estilo49" id="uleftColumn">

<p align="center"><strong>BUSCADORES</strong></p>
<br/>
<%
boolean verproyectos=false;
```

```

if(per.getNacionales().equals("administrar"))
verproyectos=true;
if(per.getNacionales().equals("ver"))
verproyectos=true;
if(per.getNacionales().equals("usuario"))
verproyectos=true;

if(per.getInternacionales().equals("administrar"))
verproyectos=true;
if(per.getInternacionales().equals("ver"))
verproyectos=true;
if(per.getInternacionales().equals("usuario"))
verproyectos=true;

if(per.getOtros_marcos().equals("administrar"))
verproyectos=true;
if(per.getOtros_marcos().equals("ver"))
verproyectos=true;
if(per.getOtros_marcos().equals("usuario"))
verproyectos=true;
if(verproyectos){
%>
<form method="post" action="Proyectos.jsp">
<p align="left">Proyectos:
<input name="Cadena" type="text" size="14" />
<input alt="Buscar" class="send-btn" name="busqueda" type="image" id="Send"
src="images/go.gif" />
<a href="busquedaAvanzadaP.jsp"></a>
<input type="hidden" name="ordenarPor" value="codigo" />
<input type="hidden" name="pagina" value="1" />
<input type="hidden" name="tipo" value="busqueda" />
</p>
</form>
<%>

if(!per.getOrganismos().equals("nada")){%>
<form method="post" action="Organismos.jsp">
<p align="left">Organismos:

```

```
<input name="Cadena" type="text" size="14" />
<input alt="Buscar" class="send-btn" name="busqueda" type="image" id="Send"
src="images/go.gif" />
<input type="hidden" name="ordenarPor" value="codigo" />
<input type="hidden" name="pagina" value="1" />
<input type="hidden" name="tipo" value="busqueda" />
</p>
</form>
<%>

if(!(per.getPersonal_docente().equals("nada"))||(!per.getPersonal_no_docente().equals("usuario
"))){%>
<form method="post" action="Investigadores.jsp">
<p align="left">Invs. en grupos de inv.:
<input name="Cadena" type="text" size="14" />
<input alt="Buscar" class="send-btn" name="busqueda" type="image" id="Send"
src="images/go.gif" />
<input type="hidden" name="ordenarPor" value="codigo" />
<input type="hidden" name="pagina" value="1" />
<input type="hidden" name="tipo" value="busqueda" />
<input type="hidden" name="Boton" value="pulsado" />
<input type="hidden" name="en_grupos" value="Si" />
</p>
</form>
<%>

if(!(per.getPersonal_docente().equals("nada"))||(!per.getPersonal_no_docente().equals(
"usuario"))){%>
<form method="post" action="Investigadores.jsp">
<p align="left">Investigadores:
<input name="Cadena" type="text" size="14" />
<input alt="Buscar" class="send-btn" name="busqueda" type="image" id="Send"
src="images/go.gif" />
<input type="hidden" name="ordenarPor" value="codigo" />
<input type="hidden" name="pagina" value="1" />
<input type="hidden" name="tipo" value="busqueda" />
<input type="hidden" name="Boton" value="pulsado" />
</p>
</form>
<%>
if(!per.getGrupos().equals("nada")){%>
```

```

        <form method="post" action="GruposOficial.jsp">
            <p align="left">Grupos Inv :
            <input name="Cadena" type="text" size="14" />
            <input alt="Buscar" class="send-btn" name="busqueda" type="image" id="busqueda"
src="images/go.gif" />
            <a href="busquedaAvanzadaG.jsp"> </a>
            <input type="hidden" name="ordenarPor" value="Alfabeticamente" />
            <input type="hidden" name="pagina" value="1" />
            <input type="hidden" name="tipo" value="busqueda" />
            <input type="hidden" name="Boton" value="pulsado" />
        </p>
    </form>
<%>
    if(!per.getGrupos().equals("nada")){%>
        <form method="post" action="actividadesgis.jsp">
            <p align="left">Actividades Grupos:
            <input name="cadena" type="text" size="14" />
                <input name="primeravez" type="hidden" value="No" />
                <input alt="Buscar" class="send-btn" name="busqueda"
type="image" id="busqueda" src="images/go.gif" />
        </p>
    </form>
    <%}%>
        <p align="left" class="Estilo50">para una b&uacute;squeda avanzada de elementos
pinchar en el m&aacute;s. </p>
        <p align="left">-----</p>
        <p align="left"><em><%=name%> </em></p>
        <p align="left"><a href="PerfilUsuario.jsp">Perfil</a></p>
        <p align="left">&nbsp;</p>
            <p align="left"><strong>Cuadros de Mando: </strong></p>
            <%
                //Permiso solo para los que puedan
                //Cuadro europeo:
                //Usuario externo no puede verlo
            <%>
            if((!per.getInternacionales().equals("nada"))&&(!per.getInternacionales().equals("usuario
"))){%>
                <p align="left" class="td1">

```

```
Europeos:
    <input type="button" size='3' vspace="22"height="8" width="12"style="cursor: pointer"
name="Cuadro2" value="Cuadro FP6" onclick="javascript:abrir()"/>
</p>
<%}
if(!per.getPlan_calidad().equals("nada")){%>

    <p align="left" class="td1">
        Grupos de inv:
            <input type="button" size='3' vspace="22"height="8" width="12"style="cursor: pointer"
name="Cuadro22" value="Plan Calidad" onclick="javascript:abrir2()"/>
        </p>
    <%}%>
    <p>-----</p>
        <p align="left"><a href="menu.jsp"> Inicio </a></p>
        <p align="left"><a href="CerrarSesion.jsp"> Desconectar </a></p>
    </form>
</div>
<div id="lleftColumn">
<%
boolean algo_administra=false;
    if(per.getInternacionales().equals("administrar"))
        algo_administra=true;
    if(per.getNacionales().equals("administrar"))
        algo_administra=true;
    if(per.getOtros_marcos().equals("administrar"))
        algo_administra=true;
    if(per.getPersonal_docente().equals("administrar"))
        algo_administra=true;
    if(per.getPersonal_no_docente().equals("administrar"))
        algo_administra=true;
    if(per.getGrupos().equals("administrar"))
        algo_administra=true;
    if(per.getOrganismos().equals("administrar"))
        algo_administra=true;

if(algo_administra){
%>

<h1>Administraci&oacute;n</h1>
```

```
<p align="left"><span class="Estilo49"><a href="panelAdministrador.jsp">Panel de
Administraci&oacute;n </a></span></p>
```

```
<p align="left" class="Estilo49"><br />
```

```
</p>
```

```
<%}%>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
-----FIN BarraLateral.jsp-----
```

Además de esta barra el sistema cuenta con otro elemento común, que es una barra superior, compuesta de un logo y un menú despegable programado en javascript.

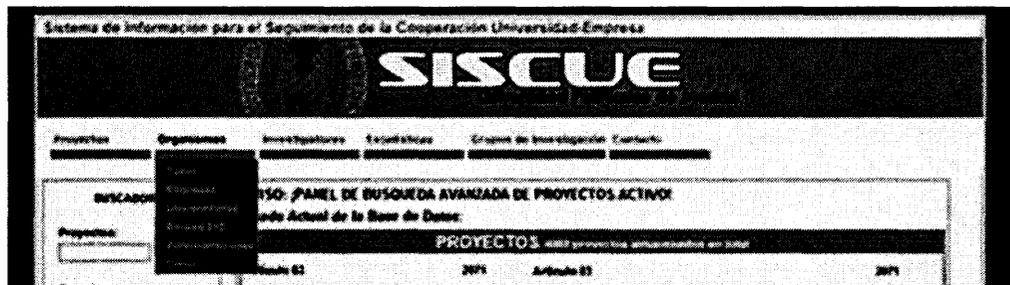


FIGURA 4-16: BARRA DE NAVEGACIÓN SUPERIOR

Dicho menú se importa de un archivo (menu.js), al igual que otros archivos javascript usados para validación de formularios o trato de datos.

Teniendo en cuenta que estos elementos son comunes a todo el sistema para facilitar su navegación, el resto de la navegación sistema podría describirse de la siguiente forma:

Las opciones de navegación de la barra lateral son:

- **Búsqueda de proyectos** por nombre
- Acceso a **búsqueda avanzada de proyectos**
- **Búsqueda de organismos** por nombre
- **Búsqueda de investigadores** por nombre
- **Búsqueda de investigadores relacionados con un grupo de investigación** por nombre
- **Búsqueda de grupos de investigación** por nombre
- Acceso a **búsqueda avanzada de grupos de investigación**
- **Búsqueda de actividades** en la memoria de investigación
- Acceso al **perfil** del usuario
- Acceso a elementos de **administración** general por parte de usuarios administradores
- Acceso al **cuadro de mando Europeo**
- Acceso al **Plan de calidad** de la investigación

- **Desconexión**
- **Ir al inicio del sistema**

Las opciones de navegación de la barra superior son:

- **Proyectos:** Todos, Europeos, Internacionales, Nacionales, Regionales, Convenios, Artículo83, Licencias, Cátedras y Título Propio.
- **Organismos:** Todos, Empresas, Universidades, Entidades I+D, Administraciones, Otros.
- **Investigadores:** Todos, Principales.
- **Estadísticas.**
- **Grupos de investigación:** Grupos y Memorias de investigación.

Aparte de estas opciones de navegación, comunes a todo el sistema, cada hoja cuenta con las suyas propias para la movilidad.

A continuación se describen el resto de hojas que completan el sistema, omitiendo todas aquellas que su labor sea intermedia o final (como, por ejemplo, la gestión del usuario que accede, no es visible, pero pasa por ella antes de acceder al sistema, autoredireccionándose). También son hojas jsp, que usan los paquetes de la aplicación descrita anteriormente para la obtención de servicios.

## Proyectos.jsp

Lista todos los proyectos elegidos filtrándolo por las opciones configuradas. Creando un link al elemento proyecto por cada proyecto encontrado (Ver screen, donde se ha buscado un proyecto por la frase "Prueba de proyecto para presentación"). También se puede acceder aquí desde la barra lateral buscando por nombre, listando todas las equivalencias encontradas.

The screenshot shows a web interface for project management. At the top right, it indicates 'Página: 1' and 'Hay 1 Proyectos'. Below this are search filters: 'Marco de Colaboración:' with a dropdown menu set to 'Todos', 'Centros:' with a dropdown menu set to 'Todos', and 'Años:' with a dropdown menu set to 'Todos'. Below the filters is a table with the following data:

Código	Título	Marco
E-00-0000-00	Prueba de proyecto para presentación de PFC de Siscue	Europeo PFP

TABLA 4-14: PROYECTOS

- Opciones de navegación:
  - Proyecto (Proyecto.jsp)

### Proyecto.jsp

Ficha del proyecto seleccionado, mostrando los datos del mismo así como de todas sus relaciones (Organismos, investigadores, centros, departamentos, ...)

**Prueba de proyecto para presentación de PFC de Siscue**

Codigo del Proyecto	E-00-0000-00
Año:	2007
Fecha Inicio	null
Fecha Entrada OTT	
Papel de la UPM	
Observaciones	
Meses Duracion	12
Ambito Geográfico	Europeo
Marco de Colaboracion	FP7
Area	Voces
Programa	IST
Instrumento	Coordinacion Action
Tipo de Proyecto	Proyecto I+D
Centro	E.U.I.T. DE TELECOMUNICACION
Departamento	INGENIERIA AUDIOVISUAL Y COMUNICACIONES
Investigador Principal	ANA GOMEZ OLIVA

**Datos Económicos:**

**2 Investigadores Relacionados:**

- ANA GOMEZ OLIVA
- Manuel Angel Jorda del Pozo

**1 Organismos Relacionados:**  
 si desea ver los contactos pulse aquí

[UPM - Universidad Politécnica de Madrid](#)  
 Universidad

FIGURA 4-17: FICHA DE PROYECTO

- Opciones de navegación:
  - Investigadores relacionados (Investigador.jsp)
  - Organismos relacionados (Organismo.jsp)
  - Modificación del Proyecto (FormularioProyecto.jsp)

## FormularioProyecto.jsp

Formulario de todo el proyecto, así como de sus relaciones, siendo todo esto modificable por un usuario con permisos para ello. Consta de dos partes, la primera de la modificación de los datos base del proyecto:

Codigo del Proyecto	E-00-0000-00
Titulo del Proyecto	Prueba de proyecto para presentación de PFC de Siscue
Año	2007
Fecha Inicio	4-12-2007
Fecha Entrada OTT	
Role de la UPM	
Observaciones	
Meses Duracion (NÚMERO ENTERO)	12
Ambito Geografico	Europeo
Marco de Colaboracion	FP7
Area	Vacio
Programa	IST
Instrumento	Coordination Action
Tipo de Proyecto	Proyecto I-D
Investigador Principal	ANA GOMEZ OLIVA
<input type="button" value="Modificar Datos"/>	

FIGURA 4-18: FORMULARIO DE PROYECTO (I)

La segunda de las relaciones existentes (agregado, borrado y modificado):

Actualmente este proyecto pertenece a:  
Centro **E.U.L.T. DE TELECOMUNICACION**  
Departamento **INGENIERIA AUDIOVISUAL Y COMUNICACIONES**  
¿Desea cambiarlo?

Centro:

---

**Modificar Investigador Principal**

---

**Asociar Investigador**

---

**Asociar Organismo**

Nombre:	<input type="text"/>	Apellidos:	<input type="text"/>
Teléfono:	<input type="text"/>	E-Mail:	<input type="text"/>
Cargo:	<input type="text"/>	Observaciones:	<input type="text"/>

EPO

---

**Lista Investigadores Asociados**

<input type="button" value="i"/> <input type="button" value="Echa"/>	ANA GOMEZ OLIVA
<input type="button" value="i"/> <input type="button" value="Echa"/>	Miguel Angel Lucas del Pozo

FIGURA 4-19: FORMULARIO DE PROYECTO (II)

- **Opciones de navegación:**
  - Investigadores relacionados (Investigador.jsp)
  - Proyecto al que se está modificando (Proyecto.jsp)
  - Organismos relacionados (Organismo.jsp)

## AgregarProyecto.jsp

Permite el agregado de un proyecto con sus diferentes relaciones.

**Agregar un Proyecto**

Título

Cuenta OTT

Año:

Fecha Inicio

Fecha Entrada OTT

Papel Upm

Meses Duración

Observaciones

Investigador Principal [vincular](#)

Ambito Geográfico:

Marco de Colaboración:

Programa:

Instrumento:

Centro:

Tipo de Proyecto

**Agregar Investigador Principal**

**Agregar Investigador Relacionado**

Investigadores Asociados

**Agregar Organismo Relacionado**

Organismos Asociados

FIGURA 4-20: AGREGADO DE PROYECTO

- Opciones de navegación:
  - Investigadores asociados (Investigador.jsp)
  - Organismos asociados (Organismo.jsp)
  - Proyecto agregado (Proyecto.jsp)
  - Formulario del proyecto agregado (FormularioProyecto.jsp)

- Volver a agregar proyecto (AgregarProyecto.jsp)

## Investigadores.jsp

Panel de búsqueda de investigadores, listándolos una vez configurado el tipo de búsqueda, también se puede acceder aquí desde la barra lateral buscando un investigador por nombre (Ver captura de pantalla, donde se ha buscado investigadores con la correspondencia "Miguel Ángel Lucas del Pozo").

Centro	Departamento	Nombre	Apellidos	
E.I.I.T. DE TELECOMUNICACION	INGENIERIA Y ARQUITECTURAS TELEMATICAS	Miguel Angel	Lucas del Pozo	Caca

FIGURA 4-21: INVESTIGADORES

- Opciones de navegación:
  - Investigadores listados (Investigador.jsp).

## Investigador.jsp

Ficha de un investigador y todas sus relaciones, como su grupo de investigación (si tiene), proyectos en los que ha participado, etc.

Miguel Angel Lucas del Pozo	
Nombre:	Miguel Angel
Apellidos:	Lucas del Pozo
Centro:	E.U.I.T. DE TELECOMUNICACION
Departamento:	INGENIERIA Y ARQUITECTURAS TELEMATICAS
DNI:	00000000
E-Mail:	prueba@prueba.com
Telefono:	911111111
Fax:	
Modo:	
Tipo de Investigador (usado en acusel):	Docente doctor
Dedicación:	Completa
Sexenios:	0.0
Doctor:	Doctor
Año lectura de tesis:	Desconocido
Grupo de Investigación:	Sin grupo de investigación
Categoría investigador:	CONT. DE INVESTIGACION UPM-OTT
Tipo de Contrato:	

**1 Proyectos Relacionados :**

Id	Numero	Descripcion	Participante
EuropeoFP7	L-00-0000-00	Prueba de proyecto para presentación de PFC de Ciencias	Participante

[Modificar Datos](#)

[Borrar Investigador](#)

FIGURA 4-22: FICHA DE INVESTIGADOR

- Opciones de navegación:
  - Proyectos relacionados (Proyecto.jsp).
  - Borrado del investigador, solo por administradores con permisos.
  - Modificado del investigador (FormularioInvestigador.jsp).
  - Grupo de investigación (Si tiene, GrupoInvestigacionOficial.jsp).

## FormularioInvestigador.jsp

Formulario del investigador para la modificación de sus datos y sus relaciones.

Nombre	Miguel Angel
Apellidos	Lucas del Pozo
Centro	E.U.I.T. DE TELECOMUNICACION
Departamento	INGENIERIA Y ARQUITECTURAS TELEMATICAS
DNI (sin la letra)	00000000
E-mail	prueba@prueba.com
Teléfono	911111111
Fax	
Móvil	
Tipo de investigador (estado en curso)	Docente doctor
Categoría investigador	CONT. DE INVESTIGACION UPM-OTT
Tipo de contrato	
Docencia	Completa
Señoras	0.0
Doctor	Doctor
Año lectivo de inicio	0
Grupo de investigación	sin grupo de investigación

Capital Proyecto	
<input type="text"/>	<input type="button" value="asociar"/>

Proyectos Relacionados			
<input type="button" value="B"/>	E-00-0000-00	Prueba de proyecto para presentación de PFC de Siscue	Participante

FIGURA 4-23: FORMULARIO DE INVESTIGADOR

- Opciones de navegación:
  - Proyectos relacionados (Proyecto.jsp).
  - Borrado del investigador, solo por administradores con permisos.
  - Grupo de investigación (Si tiene, GrupoInvestigacionOficial.jsp).

## AgregarInvestigador.jsp

Ficha de un investigador para darle de alta en el sistema (Con los permisos necesarios).

Agregar Investigador	
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
DNI (Sin letra)	<input type="text"/>
E-Mail	<input type="text"/>
Teléfono	<input type="text"/>
Fax	<input type="text"/>
Tipo de Investigador	<input type="text" value="sin tipo asignado"/>
Dedicación	<input type="text" value="Desconocida"/>
Doctor	<input type="text" value="No se sabe"/>
Centro	<input type="text" value="NO ADSCRITOS A CENTRO"/>
Categoría Investigador	<input type="text" value="SIN CATEGORIA"/>
Tipo de contrato	<input type="text"/>

**NOTA:** Una vez agregados los datos básicos del investigador, podrá completar la información que no aparece aquí, y relacionarlo con grupos de investigación, proyectos, etc... si tiene los permisos necesarios.

FIGURA 4-24: AGREGADO DE INVESTIGADOR

- **Opciones de navegación:**
  - Investigador agregado (Investigador.jsp)
  - Formulario de investigador agregado (FormularioInvestigador.jsp)
  - Volver a agregar investigador (AgregarInvestigador.jsp)

## Organismos.jsp

Ficha de un organismo y todas sus relaciones, incluidas en las relaciones con proyectos los contactos que tienen que ver en esa participación.

**Organismo de prueba**

Cif	00000000
CNAE	
País	España
Web	www.prueba.com
E-Mail	prueba@prueba.es
Teléfono	91000000
Fax	
Calle	C Pallas de Madrid nº7
Código Postal	28040
Ciudad	Madrid
Provincia	Madrid
Comunidad	Madrid (Comunidad de)
Tipo Organismo:	Entidad I+D: Centro de Innovación y Tecnología (Registrado en Madrid)

**1 Proyectos Relacionados:**

si desea ocultar los contactos pulse [aquí](#)

[Prueba de proyecto para presentación de PFI de Cáceres](#)

**Código OTT: E-00-0000-00**

**Europeo: FPI**

<b>Nombre:</b>	Miguel Angel	<b>Apellidos:</b>	Lucas del Pozo
<b>Teléfono:</b>	91000000	<b>E-Mail:</b>	
<b>Cargo:</b>		<b>Observaciones:</b>	

FIGURA 4-25: FICHA DE ORGANISMO

- **Opciones de navegación:**
  - Proyectos relacionados (Proyecto.jsp)
  - Borrado del organismo, solo por administradores con permisos.
  - Modificado del organismo (FormularioOrganismo.jsp).

## FormularioOrganismo.jsp

Formulario del organismo seleccionado para la modificación de todos sus parámetros y de sus relaciones.

Nombre Organización	Organismo de prueba
Cif	00000000
CHAE	
País	España
Web	www.prueba.com
E-Mail	prueba@prueba.es
Teléfono	91000000
Fax	
Calle	C Ramiro de maetzu nº7
Código Postal	28040
Ciudad	Madrid
Provincia	Madrid
Comunidad	Madrid (Comunidad de)

Actualmente este organismo es Entidades I+D : Centro de Innovación y Tecnología (Registrado en Mec)

Entidades I+D  Centro de Innovación y Tecnología (Registrado en Mec)

Asociados			
<input type="text"/>	<input type="button" value="ASOCIAR"/>		
Nombre:	<input type="text"/>	Apellidos:	<input type="text"/>
Teléfono:	<input type="text"/>	E-Mail:	<input type="text"/>
Cargo:	<input type="text"/>	Observaciones:	<input type="text"/>
EPO	<input type="checkbox"/>		

FIGURA 4-26: FORMULARIO DE ORGANISMO

- Opciones de navegación:
  - Proyectos relacionados (Proyecto.jsp)
  - Organismo que estamos modificando (Organismo.jsp)
  - Borrado del Organismo

## AgregarOrganismo.jsp

Desde la parte administradora podemos acceder (Siempre con los permisos necesarios) al agregado de un organismo nuevo.

Agregar Organismo	
Nombre Organismo	Organismo de prueba
País	España
CIF	00000000
CHAE	
Teléfono	91000000
Fax	
Web	www.prueba.com
e-mail	prueba@prueba.es
Calle	C/ Ramiro de maetzu nº7
Código Postal	28040
Ciudad	Madrid
Provincia	Madrid
Comunidad	Madrid Comunidad de
Tipo	Entidades I-D
Especificación tipo	Centro de Innovación y Tecnología (Registrado en Mec)
<input type="button" value="Agregar"/>	

FIGURA 4-27: AGREGADO DE ORGANISMO

- Opciones de navegación:
  - Organismo agregado (Organismo.jsp)
  - Volver a agregar organismos (AgregarOrganismo.jsp)
  - Formulario del organismo agregado (FormularioOrganismo.jsp)

## GruposInvestigacion.jsp

Lista todos los grupos de investigación configurados por diferentes filtros (Estado del grupo, líneas que tiene, áreas de conocimiento a las que pertenecen, y palabras clave en el nombre del grupo). En la captura se ha buscado grupos que en el nombre tengan la palabra "Prueba".

Centro	Grupo de investigación	Tamaño	Inv. Responsable
E S I T DE TELECOMUNICACION	En estado: Grupo de investigación de prueba de...	1	Módulo: Grupo de prueba de...

FIGURA 4-28: GRUPOS DE INVESTIGACIÓN

- **Opciones de navegación:**
  - Grupo de investigación (GrupInvestigacionOficial.jsp)
  - Investigador responsable del grupo (Investigador.jsp)

### GrupoInvestigacionOficial.jsp

Ficha del grupo de investigación reconocido. Tiene tanto la información propia del grupo, como las relaciones con los investigadores que pertenecen a dicho grupo, líneas, áreas, etc.

Grupo de investigación de prueba Siscue	
Investigador Responsable:	Ingen. Angel Lopez de Puga
Centro:	E.U.I.T. DE TELECOMUNICACION
Otros Centros implicados:	
Web:	www.gruposiscue.es
E-mail:	
Telefono:	9150000
Fax:	
Convocatoria:	P
Código OTI:	0-00-0000-00
Acónimo:	GPS
ESTADO DEL GRUPO:	En proceso de consolidación
Observaciones:	Null
Tamaño de grupo → 127 DATOS	
Miembros de grupo:	1
Total personal:	1
Fecha de registro:	Desconocida
Número de registro:	
Memorias:	
Ayudas solicitadas:	

FIGURA 4-29: FICHA DE GRUPO (I)

Investigadores asociados activos				
<i>Nota: Los proyectos que se visualizan por investigador son de fuentes oficiales (Datos cargados en el Siscue)</i>				
CONT. DE INVESTIGACION UPM-OTT				
Nombre	DNI	Fecha Alta	Fecha Modificación	
<input checked="" type="checkbox"/> Miguel Ángel Lucas del Pozo	00000000	2007-12-05	desconocida	
Historial Investigadores (De baja)				
<i>Nota: Los proyectos que se visualizan por investigador son de fuentes oficiales (Datos cargados en el Siscue)</i>				
Áreas de conocimiento				
<input checked="" type="checkbox"/> Ingeniería Telemática				
Líneas de Investigación				
Linea	Investigador responsable			
Desarrollo de sistemas de información	Miguel Ángel Lucas del Pozo			
Ayudas del Grupo				
Convocatoria	Referencia	Título ayuda	Cantidad Concedida	Fecha ayuda

FIGURA 4-30: FICHA DE GRUPO (II)

- **Opciones de navegación:**

- Investigadores relacionados (Investigador.jsp)
- Proyectos relacionados con los investigadores (Proyecto.jsp)
- Líneas de investigación (si tienen información adicional) (Linea.jsp)
- Memorias de investigación relacionadas con el grupo (Actividadesgi.jsp)
- Grupos de investigación en el mismo área de conocimiento (GrupoInvestigacionOficial.jsp)

## FormularioGrupoInvestigacionOficial.jsp

Formulario del Grupo de investigación seleccionado para la modificación de todos sus parámetros y de sus relaciones.

Modificar el Grupo

### Modificación de grupo de investigación

Nombre Grupo:	Grupo de investigación de prueba Siacue
Investigador Responsable:	Miguel Angel Lucas del Pozo
Web:	www.gruperprueba.es
Email:	
Teléfono:	9100000
Fax:	
Convocatoria:	<input checked="" type="checkbox"/>
Código OTT:	0-00-0000-00
Apónimo:	GIPS
Estado:	En proceso de consolidación
Observaciones:	null
Número de Registro:	
Fecha de registro:	<input type="text"/>

Ayudas solicitadas:

Agregar petición de ayuda    Año de actividades: 2006    Concedidas: Desconocido   

FIGURA 4-31: FORMULARIO DE GRUPO (I)

Agregar Ayuda

Agregar Investigador

**Investigadores asociados activos**

Nota: Para modificar las fechas o categorías debe hacerlo uno a uno y dar al botón de "modificar" correspondiente

<input checked="" type="checkbox"/> desactivar	Miguel Angel Lucas del Pozo	Fecha Alta:	[CONT. DE INVESTIGACION UPMOTT]
	<input type="button" value="fecha"/>	<input type="text" value="05-12-2007"/>	<input type="button" value="modificar"/>

**Historial Investigadores del grupo (De baja)**

Nota: Para modificar las fechas o categorías debe hacerlo uno a uno y dar al botón de "modificar" correspondiente

Agregar Area: [Actividad Fisica y Deposte]

Areas ya asociadas:

- Ingeniería Telemática

Agregar Línea:

Líneas ya asociadas:

Borrar	Línea	Categoría	Investigador responsable
<input checked="" type="checkbox"/>	Desarrollo de sistemas de Información	[FICHA]	Miguel Angel Lucas del Pozo <input type="button" value="modificar"/>

Este grupo no tiene campos adicionales

FIGURA 4-32: FORMULARIO DE GRUPO (II)

- **Opciones de navegación:**
  - Grupo de investigación que estamos modificando (GrupInvestigacionOficial.jsp)
  - Formularios de líneas de investigación (FormularioLinea.jsp)
  - Gestión de ayudas a grupos (FormularioAyuda.jsp)
  - Investigadores relacionados (Investigador.jsp)

## Actividadesgis.jsp

Es un panel de búsqueda de actividades sacadas de las memorias de investigación de los grupos. Nos permite buscar actividades filtrándolos por tipo de actividad, palabras clave, centros de la UPM con lo que está relacionado y año de la actividad. El tipo de actividad se disgrega en dos niveles, los primeros son los epígrafes usados como indicadores en el Plan General de Calidad de la Investigación, y los segundos el tipo de actividad dentro de ellos:

- Proyectos I+D+i:
  - Proyectos
- Formación de investigadores:
  - Tesis
  - Becarios y contratados para la realización del Doctorado
  - Personal investigador contratado
  - Estancias y sabáticos
- Difusión de resultados
  - Artículos
  - Libros
  - Capítulos de libros
  - Ponencias
  - Informes para las aapp
  - Conferencias
  - Otras publicaciones
- Explotación de resultados
  - Patentes
  - Empresas
  - Registros de software
  - Otros méritos
  - Responsabilidades
  - Premios

Una vez seleccionado todo lo que queremos sobre las actividades, nos lista todas las encontradas con esos filtros.

CONFIGURACIÓN DE LA BÚSQUEDA			
Actividad por Palabra clave:	<input type="text"/>	Ordenar por:	Per nombre de act
Tipo:	Todo	Grupo que tenga línea con la palabra:	<input type="text"/>
Año de la actividad en el grupo:	2006		
Centro:	Todos		
<b>BUSCAR</b>			
Cantidad a mostrar: 100	<input type="button" value="Cambiar"/>	Páginas: 1	Se muestran del: 1 al 0
			Página: 1
ACTIVIDADES DE INVESTIGACIÓN			
Ítems	Tipo	Actividad	Memoria de Investigación

FIGURA 4-33: BUSQUEDA DE ACTIVIDADES DE INVESTIGACIÓN

- Opciones de navegación:
  - Actividad encontrada (Pub.jsp)
  - Memoria de investigación del grupo al que pertenece (Actividadesgi.jsp)

## Actividadesgi.jsp

Atendiendo a la estructura de tipos de actividades vistos anteriormente, esta página muestra todo con detalle sobre las actividades del grupo relacionado en un año seleccionado, pasando por encima el ratón se nos abre un resumen (JavaScript) de dicha actividad. En la captura no se muestran los datos resultantes debido a la posible confidencialidad de algunos de ellos.

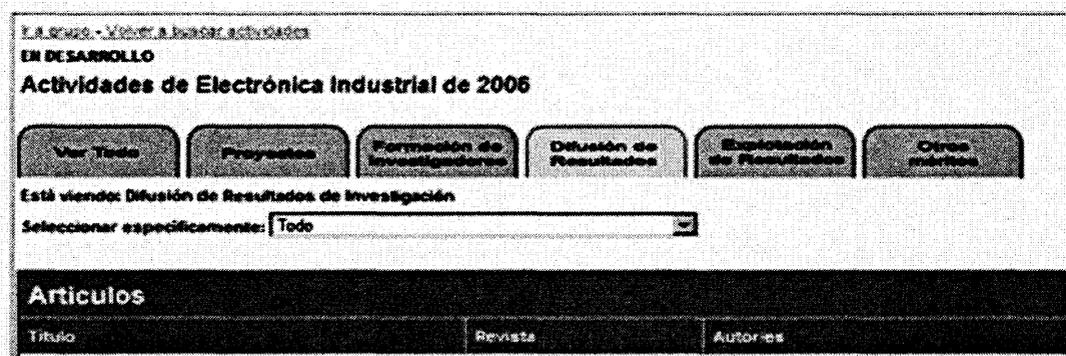


FIGURA 4-34: MEMORIAS DE INVESTIGACIÓN

- Opciones de navegación:
  - Fichas de actividades (Actividadesgi.jsp)
  - Grupo al que pertenecen las actividades (GrupoInvestigacionOficial.jsp)
  - Panel de búsqueda de actividades (actividadesgis.jsp)

## Pub.jsp

Ficha de una actividad de investigación relacionada con una memoria.

**NAVEGACION:** panel de actividades - Resto de actividades de esta memoria

**Actividad perteneciente a:**

**Año: 2006**

Grupo de investigación:

Proyecto	
Título de Actividad:	
Tipo de proyecto	<b>Contratos y convenios de financiación privada</b>
Entidad financiera	
Año de concesión	<b>2007</b>
descripcion	

Participantes	
director	
participante	
participante	

FIGURA 4-35: FICHA DE ACTIVIDAD DE INVESTIGACIÓN

- Opciones de navegación:
  - Panel de actividades (actividadesgis.jsp)
  - Resto de actividades de la memoria (Actividadesgsi.jsp)
  - Grupo de investigación relacionado con la actividad (GrupoInvestigacionOficial.jsp)



Nombre del grupo		TOTAL: 3.88679 % del total
Comisiones de sucesos académicos		TOTAL: 13.82224862081
Comisión representada por proyectos y convenios de convocatorias públicas competitivas	4.47101	Abstrato: 1.207.207.9
Comisión representada por proyectos y convenios de financiación pública no competitiva	0.32086	Abstrato: 1.601.4
Comisión representada por convenios y convenios de financiación privada	9.45145	Abstrato: 6.104.820.6
Otros ingresos	0.68027	Abstrato: 2.400.6
Formación de investigadores y residentes		TOTAL: 1.1071428714287
Beas doctorales propias por según PDI de la unidad	0.22871	Abstrato: 1.8
Beas de contrato para la realización del Doctorado	0.0	Abstrato: 0.0
Contratos de personal investigador	1.27142	Abstrato: 41.8
Becas y ayudas recibidas en programas oficiales	0.0	Abstrato: 0.0
Difusión de resultados de investigación		TOTAL: 0.847058623254118
Límites e capítulos en libros publicados en el ámbito	0.22641	Abstrato: 4.8
Publicaciones en revistas internacionales incluidas en el SCi con factor de impacto alto (más superior)	0.10302	Abstrato: 0.0
Publicaciones en revistas internacionales incluidas en el SCi con menor factor de impacto (más inferior)	0.34112	Abstrato: 2.8
Publicaciones en otras revistas del momento	0.32238	Abstrato: 3.8
Participación en congresos internacionales con referencias	0.33388	Abstrato: 19.8
Participación en congresos nacionales con referencias	0.21427	Abstrato: 4.8
Informes para los AAPP o sus organismos dependientes	0.0	Abstrato: 0.0
Conferencias invitadas en congresos internacionales	0.02238	Abstrato: 1.8
Exposición de resultados y transferencia de tecnología		TOTAL: 0.3821
Patentes del sistema internacional en explotación	0.0	Abstrato: 0.0
Patentes con sistema internacional en explotación	0.0	Abstrato: 0.0
Patentes de ámbito nacional en explotación	0.0	Abstrato: 0.0
Patentes de ámbito nacional en explotación	0.0	Abstrato: 0.0
Registros de software en explotación	0.3821	Abstrato: 0.0
Depositos creativos	0.0	Abstrato: 0.0
Reconocimiento de medallas		TOTAL: 0.1
Adquisición de medallas (Premios Nacionales y Premios Resúmenes)	0.1	Abstrato: 0.0

FIGURA 4-38: RESULTADO DEL PLAN DE CALIDAD

También podemos calcular los valores absolutos de la suma de actividad de todos los grupos (por ejemplo toda la suma económica), o las ayudas a grupos configurando unos parámetros de financiación.

- Opciones de navegación:
  - Calculo de financiación (PlanCalidadResultado.jsp)
  - Memoria de los grupos de investigación (Actividadesgi.jsp)

- Excel de todo
- Valores absolutos (Absolutos.jsp)
- Gráfica del reparto de pesos

## Centro.jsp

Panel de generación de consultas estadísticas de proyectos configurables, podemos elegir en torno a que queremos hacer las consultas (investigadores, organismos, grupos de investigación, centros, departamentos), configurarlo (por todo lo relacionado con los proyectos y en torno a que hemos elegido) y seleccionar la forma de mostrar los datos (Gráficas o tablas)

The screenshot shows a web form for configuring a query. It includes several dropdown menus and a radio button selection.

**Consulta en torno:** Centros  
**Tipo de Proyecto:** Europeo  
**Marcos de Colaboración:** FP6  
**Años Proyecto:** 2003-2006  
**Presupuesto:** Todos

**Tablas**   
**Gráficas**

**HA SELECCIONADO CENTROS**

**Tipo de Consulta:** Número de Proyectos  
**Ordenar por:** Consulta de mayor a menor

**Se mostrará el resultado en Gráficas**

**Tipo de gráficas:** Queso  
**Número de Valores\***

\*Si el número supera el máximo de la consulta, sacará dicho máximo, en blanco coge por defecto 10 valores

Below the form is a circular button with a globe icon.

FIGURA 4-39: CONFIGURACIÓN DE CONSULTA SENCILLA

- Opciones de navegación:
  - Resultado de la consulta (Consulta.jsp)

## Consulta.jsp

Desde el panel de generación, una vez configurada la consulta, nos muestra los resultados en el formato que le hallamos pedido (Gráfica o tablas) Permitiendo exportar esos resultados a Excel o Pdf. En este caso se pidió ver la participación 2003-2006 por Centros UPM en el FP6 Europeo, y mostrarlo en queso.

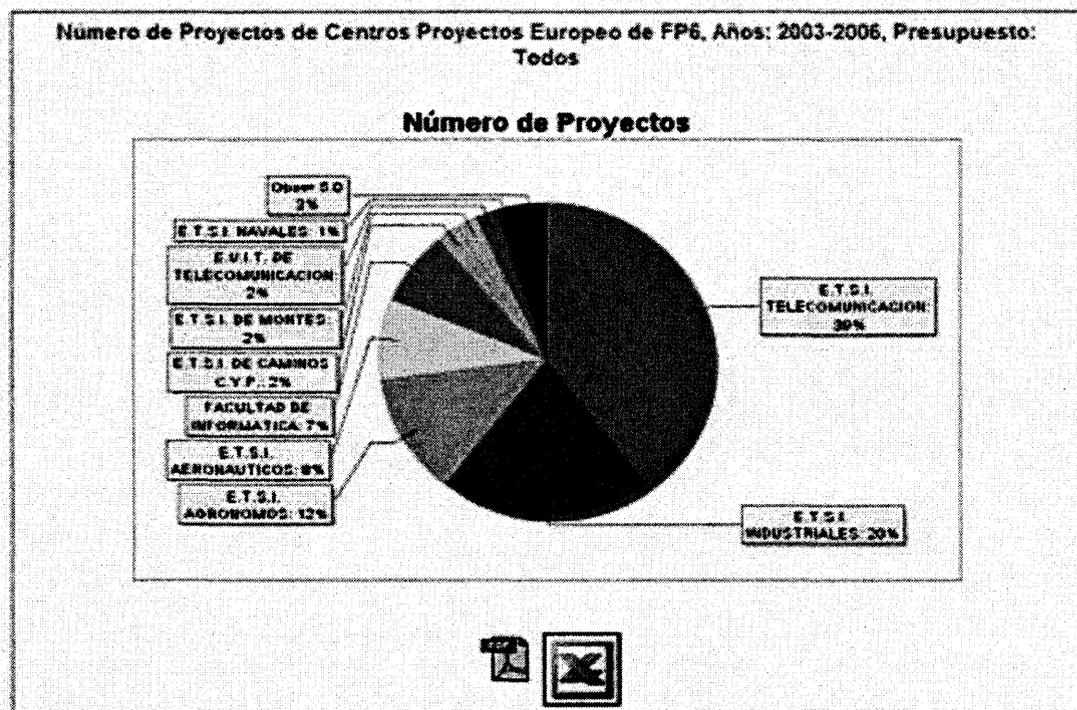


FIGURA 4-40: EJEMPLO DE RESULTADO DE CONSULTA SENCILLA

## BusquedaAvanzadaP

Basándonos en una hoja de este tipo, se configura qué tipo de consulta avanzada queremos hacer al sistema sobre proyectos.

Para ello podemos elegir que parámetros queremos del proyecto, que datos de sus relaciones, y que queremos que no saque (ara más información, ver la guía de usuario)

**Búsqueda Avanzada de Proyectos**

*Palabra Clave:*

*Marco de colaboración:* Todos

*Marco específico:* Todos

*Años Proyectos:* 2003-2006

*Presupuesto:* Todos



**POLITÉCNICA**

**Datos a mostrar del proyecto:**

<input type="checkbox"/> Título	<input type="checkbox"/> Tipo de Proyecto (I+D, formación,...)
<input type="checkbox"/> Código OTT	<input type="checkbox"/> Centro
<input type="checkbox"/> Investigador Principal	<input type="checkbox"/> Departamento
<input type="checkbox"/> Marco de Colaboración (Europeo,Nacional,Art 83..)	<input type="checkbox"/> Presupuesto concedido UPM
<input type="checkbox"/> Submarco (Europeo: FP6,EUREKA...)	<input type="checkbox"/> Programa
<input type="checkbox"/> Año	<input type="checkbox"/> Instrumento
<input type="checkbox"/> Observaciones	<input type="checkbox"/> Área de conocimiento

FIGURA 4-41: CONFIGURACIÓN DE BUSQUEDA AVANZADA DE PROYECTOS (I)

Buscar sólo los proyectos que tengan relación al menos con los siguientes organismos:

Tipo:

Nacionalidad:

↳ País:

Datos a mostrar del organismo:

Nombre Organismo  CIF

Tipo de organismo (General)  País

Tipo de organismo (Específico)  Datos de contacto (Email, teléfono, dirección,...)

NOTA: Si selecciona algún campo a mostrar de organismos relacionados, el sistema buscará todas las relaciones con los organismos deseados (existan o no) y mostrará los campos seleccionados, pudiéndose ver proyectos repetidos para mostrar las diferentes relaciones con todos los organismos asociados.

Buscar sólo los proyectos que pertenecen a:

Centro:

NOTA: Debido a la posibilidad de crear consultas de tamaño considerable, no se pueden insertar links de los elementos a sus fichas.

Exportar directamente a Excel

FIGURA 4-42: CONFIGURACIÓN DE BUSQUEDA AVANZADA DE PROYECTOS (II)

Una vez configurada la consulta, podemos exportarla a Excel directamente o sacar la tabla con los resultados deseados, (Que también podremos llevarnos a excel), la tabla nos la saca en la hoja CreacionBusquedaP.jsp y queda de la siguiente forma:

Orden	Título	Tipo	Centro	Investigador/Responsable	Año	NombreOrganismo	TipoOrganismo
1	Proyectos de Investigación	Prueba y Ensayo	ETS I AGRICOLAS	...	2006	...	Empresa
2	...	Prueba y Ensayo	ETS DE ARQUITECTURA	...	2005	...	Empresa
3	...	Proyecto I+D	FACULTAD DE INFORMATICA	...	2007	...	Empresa
4	...	Apoyo Tecnológico y Asesoría	ETS I TELECOMUNICACION	...	2003	...	Empresa

FIGURA 4-43: TABLA DE BUSQUEDA AVANZADA DE PROYECTOS

## BusquedaAvanzadaG

Al igual que en proyectos, la búsqueda avanzada de grupos tiene la misma estructura pero siendo sobre grupos de investigación, permitiéndonos elegir los parámetros de los grupos y que queremos que nos muestre en el resultado.

**Búsqueda Avanzada de Grupos de Investigación**



**POLITÉCNICA**

**Palabra Clave:**

**Estado:**

**Convocatoria:**

**Datos a mostrar del Grupo de Investigación:**

<input checked="" type="checkbox"/> Nombre	<input type="checkbox"/> Observaciones
<input checked="" type="checkbox"/> Investigador Responsable	<input checked="" type="checkbox"/> Centro
<input type="checkbox"/> Otros Centros Implicados	<input type="checkbox"/> E-mail IR
<input type="checkbox"/> Acrónimo	<input type="checkbox"/> E-mail Grupo
<input type="checkbox"/> Convocatoria	<input type="checkbox"/> Teléfono
<input type="checkbox"/> Estado	<input type="checkbox"/> Fax
<input type="checkbox"/> Código OTT	<input type="checkbox"/> Web

FIGURA 4-44: CONFIGURACIÓN DE BUSQUEDA AVANZADA DE GRUPOS (I)

También podemos ir configurando todo lo que queremos sobre los elementos relacionados con los grupos de investigación (líneas, áreas de conocimiento, ayudas a los grupos, personal, etc.) diciéndole qué tipo de filtro queremos, y qué datos nos debe devolver.

NOTA: Si marca la opción de Areas o Lineas, buscará las relaciones por la palabra clave insertada (Todas las relaciones si la palabra clave está en blanco) mostrando grupos repetidos para cada relación.

Areas de Investigación  
Palabra clave:

Lineas de Investigación  
 Ver Responsables de las Lineas  
Palabra clave:

Ayudas:  
 Mostrar solo grupos que tienen Ayudas  
Convocatoria:  Recuerde que si elige alguna categoría en especial de las ayudas a grupos, el Sistema sacará solo los grupos que tienen ayudas en esa convocatoria.

Datos a mostrar de las Ayudas:

<input type="checkbox"/> Título ayuda	<input type="checkbox"/> Referencia
<input type="checkbox"/> Cantidad concedida	<input type="checkbox"/> Concedido gastos ejecución
<input type="checkbox"/> Concedido becas	<input type="checkbox"/> Total importe seleccionado
<input type="checkbox"/> Convocatoria de la Ayuda	

NOTA: Si selecciona algún campo a mostrar de ayudas relacionadas, el sistema buscará todas las relaciones con las ayudas deseadas (existan o no) y mostrará los campos seleccionados, pudiéndose ver grupos repetidos para mostrar las diferentes relaciones con todas las ayudas asociadas.

FIGURA 4-45: CONFIGURACIÓN DE BUSQUEDA AVANZADA DE GRUPOS (II)

Personal del grupo de investigación:

Categorías:

Doctor:

Datos a mostrar del personal:

<input type="checkbox"/> Nombre	<input type="checkbox"/> Apellidos
<input type="checkbox"/> Categoría	<input type="checkbox"/> Email
<input type="checkbox"/> Teléfono	<input type="checkbox"/> Centro
<input type="checkbox"/> Departamento	<input type="checkbox"/> DNI
<input type="checkbox"/> Es doctor	

NOTA: Si selecciona algún campo a mostrar de personal relacionado, el sistema buscará todas las relaciones y mostrará los campos seleccionados, pudiéndose ver grupos repetidos para mostrar las diferentes relaciones con todo su personal.

Buscar sólo los grupos que pertenecen a:

Centro:   Departamento:

NOTA: Debido a la posibilidad de crear consultas de tamaño considerable, no se pueden insertar links de los elementos a sus fichas.

Exportar directamente a Excel

FIGURA 4-46: CONFIGURACIÓN DE BUSQUEDA AVANZADA DE GRUPOS (II)

Una vez configurada la consulta, el paquete ConsultasV2 recibe toda la configuración, tratándola para generar consultas cruzadas y acabar devolviendo todos los resultados que se muestran a continuación:

Orden	Link	Nombre_Grupo_Investigacion	Investigador_Principal	Centro_Grupo	Linea
1	[Link]	Electromagnético computacional aplicado a entornos y dispositivos (ECAD)	ANA ZARATA PEREZ	E.T.S.I. TELECOMUNICACION	Estudio de técnicas numéricas para el análisis y diseño de circuitos pasivos de microondas y antenas
2	[Link]	Electromagnético computacional aplicado a entornos y dispositivos (ECAD)	ANA ZARATA PEREZ	E.T.S.I. TELECOMUNICACION	Diseño de Circuitos pasivos de microondas y antenas
3	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Comunicaciones Visuales y Codificación de vídeo
4	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Video avistado
5	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Sistemas de telecomunicación y transmisión
6	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Procesos digitales de vídeo
7	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Sistemas de gestión e identificación de contenidos audio-visuals
8	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Aplicaciones audiovisuales en Densidad e Inteligencia
9	[Link]	Grupos de Aplicación de Telecomunicaciones Visuales (GATV)	JOSE MANUEL MENENDEZ GARCIA	E.T.S.I. TELECOMUNICACION	Aplicaciones de gestión de señal y comunicaciones en Sistemas Integrados de Transporte
10	[Link]	Grupos de Aplicaciones del Procesado de Señal (GAPS)	FRANCISCO JAVIER CASASUS QUINON	E.T.S.I. TELECOMUNICACION	Redes de comunicaciones digitales
11	[Link]	Grupos de Aplicaciones del Procesado de Señal (GAPS)	FRANCISCO JAVIER CASASUS QUINON	E.T.S.I. TELECOMUNICACION	Sistemas de banda estrecha

FIGURA 4-47: TABLA DE BUSQUEDA AVANZADA DE GRUPOS

Debido a su complejidad y su enorme dinamismo, no se ha abarcado por ahora la posibilidad de generar diferentes tipos de gráficas. ConsultasV2 está orientado a buscar todo lo imaginable sobre los elementos de la base de datos, construyendo enormes tablas, y por esta razón la generación automática de gráficas puede resultar ilógica. Para suplir esto, se han integrado en los cuadros de mando del interfaz Web, diferentes tipos de consultas más lógicas de las cuales sí se generan gráficas.

### 4.3.2.2 Sistema de permisos

Para gestionar la entrada de usuarios y los permisos de los mismos en el sistema, se usó un sistema sencillo de permisos. Para ello se crearon dos tablas en la propia bdd, una de usuarios, y otra de permisos, la estructura de las tablas es la siguiente:

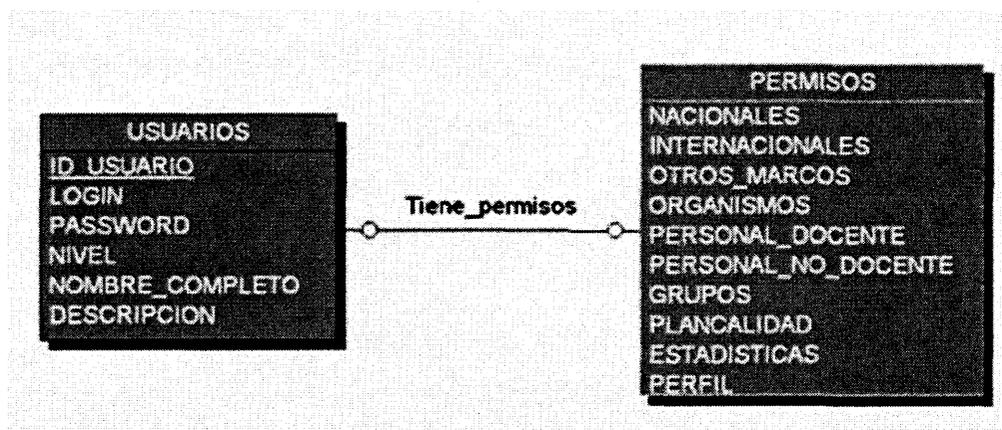


FIGURA 4-48: TABLAS DE PERMISOS Y USUARIOS

#### USUARIO:

- **Login:** Nombre de usuario (no modificable por el usuario)
- **Password:** Contraseña usada para el acceso al sistema, se guarda encriptada y en la comunicación cliente (navegador web) y servidor se pasa por SSL. (modificable por el usuario).
- **Nivel:** de 1-3, es el nivel de usuario usado para acceder a ciertos rincones de la aplicación (no modificable por el usuario).
- **Nombre\_completo:** Nombre de la persona o entidad que posee el usuario (Ej: Miguel Angel Lucas, o Oficina de Proyectos Europeos). (Modificable por el usuario).
- **Descripcion:** Breve descripción del usuario (modificable por el usuario).

#### PERMISOS:

Ningún permiso es modificable por el usuario, y cada uno es un "Enum" de cuatro posibles valores:

- **“nada”**: Permisos denegados en el bloque que tenga este valor.
- **“usuario”**: Nivel más bajo, solo puede acceder a la información pública de los elementos en los cuales tiene permiso.
- **“ver”**: es como un usuario avanzado, no puede modificar, agregar ni eliminar, pero tiene acceso a toda la información del bloque que lleve este valor.
- **“administrador”**: No solo puede navegar por todo, sino modificar, agregar y eliminar.

Los bloques sobre los que se aplican estos 4 valores son:

- **Nacionales**: Es un bloque de los proyectos de ámbito nacional
- **Internacionales**: Bloque de proyectos internacionales
- **Otros\_marcos**: Resto de proyectos
- **Personal\_docente**: Personal que pertenece a la UPM.
- **Personal\_no\_docente**: Personal externo a la UPM.
- **Grupos**: Relativo a los grupos de investigación.
- **Plan\_calidad**: Relativo al Plan de Calidad de la investigación.
- **Estadísticas**
- **Perfil**: a su propio perfil.

#### **Ejemplo de configuración de permisos:**

Creamos un usuario llamado Siscue, y le asignamos los siguientes permisos:

- **Nacionales**: ver
- **Internacionales**: ver
- **Otros\_marcos**: administrar
- **Personal\_docente**: ver
- **Personal\_no\_docente**: administrar
- **Grupos**: usuario
- **Plan\_calidad**: nada.

- **Estadísticas:** usuario
- **Perfil:** usuario

Cuando este usuario acceda al sistema, la plataforma filtra la información de tal modo que esta persona podrá modificar información relativa a proyectos que no sean ni de ámbito nacional ni internacional (por ejemplo, una cátedra), podrá administrar personal que no pertenezca a la UPM, no sabrá de la existencia del Plan de Calidad ya que no saldrán ni los links relativos a ello, y podrá navegar por la información no confidencial en estadísticas y su perfil.

#### ***4.4 Resumen de elementos usados para la creación y mantenimiento del sistema***

Para la creación y mantenimiento del sistema se ha usado todo lo siguiente.

##### **Hardware:**

- Equipo servidor DELL PowerEdge 840.
- SAI para servidor.
- Conexión a la red de servidores del rectorado UPM.

##### **Software:**

- Sistema operativo Windows 2003 Server.
- Control remoto servidor para mantenimiento.
- Java JRE 1.5: Para entorno de programación java.
- MySQL Server 5.0: Para servicio base de datos.
- Apache Tomcat 5: Para servicio web.
- Dreamweaver: Para diseño y programación interfaz web.
- Eclipse 3.3: Para programación de aplicación.
- PoweDesigner: Para creación del modelo de datos conceptual y físico de la base de datos.

# CAPÍTULO V

## RESULTADOS LOGRADOS A PARTIR DE SISCUE, CONCLUSIONES

---

### **5.1 Realización del análisis**

Como consecuencia del desarrollo de SISCUE como medio para completar el análisis de la cooperación de la Universidad Politécnica con el Sector Empresarial”, se pudieron evaluar distintas estadísticas a nivel general mostrando las relaciones de la UPM con diferentes organismos.

La posibilidad de cruzar datos nos permitía comparar tanto a nivel de participación como económico distintos tipos de proyectos tales como los realizados al amparo del Artículo 83 o los realizados en el ámbito Europeo.

Como resultado final se tuvo un informe elaborado a partir de los datos sacados del análisis (**Libro:** Análisis de la Cooperación de la Universidad Politécnica de Madrid con el Sector Empresarial, **autor:** Gonzalo León), donde se puede ver la evolución 2003-2006 de todos los marcos de colaboración donde participa la UPM. Gracias a esta elección de manipulación de datos el estudio fue bastante exhaustivo, dando mucho detalle al marco Europeo del que se creó una oficina para el fomento de la UPM en el ámbito europeo (OPE: Oficina de Proyectos Europeos).

### **5.2 Ventajas del uso de un Sistema de Información**

Como se citaba anteriormente, al usar un sistema de información se pudo ir “navegando” por todo tipo de consultas, es decir, no se cerró el análisis a un tipo predefinido, sino que usando el sistema, se fueron percibiendo diferentes conceptos no apreciados hasta el momento. Esto se hizo cruzando datos, usando tanto los paneles del interfaz web, como consultas más complejas a través del interfaz de MySQL.

La posibilidad de crear todo tipo de consultas no fue la única ventaja conseguida debida al sistema, sino que se vió que la gestión de los datos era mucho más controlada, centralizada y segura, de tal forma que si este tipo de datos era gestionado a través del sistema, permitía una realimentación de las consultas y un control de los datos más óptimo.

### **5.3 Conclusiones, un amplio abanico de posibilidades**

Recordando que el desarrollo del SISCUE parte de la necesidad de un análisis de información, una vez terminado éste, nos damos cuenta de que el esfuerzo de crear este sistema no se queda solo en dicho análisis, sino que se puede reutilizar no solo para más análisis futuros, sino además:

- Gestión efectiva de los datos relativos a la investigación.
- Creación de otras plataformas que se aprovechen de la información con la que trabaja el SISCUE, como un observatorio de investigación, mostrando los datos públicos a todo usuario que necesite navegar por la investigación de la UPM.
- Centralización de información de diferentes sectores, como por ejemplo:
  - Datos del personal docente gestionados por el Rectorado de la UPM que sean los mismos que los usados por el sistema dedicado a la investigación.
  - Datos económicos usados por la USRE o la OTT aprovechados por el sistema para la investigación.
  - Memorias de investigación sacadas de la suma de las memorias de cada investigador, donde cada uno aporta toda su actividad investigadora.

Es decir, partiendo de este sistema como comienzo de un sistema global de la investigación de la UPM, se puede ampliar intentando abarcar todo lo relativo a dicha investigación, no solo los proyectos, investigadores, grupos de

investigación y organismos, sino toda la actividad investigadora, tipos de participaciones, etc.

No hay que olvidar las ventajas del uso de un sistema de estas características para la gestión de los datos por los organismos administrativos, además de permitir el análisis sobre toda la investigación en tiempo real. Si ampliamos el sistema para el uso total por parte de los organismos administrativos, nos damos cuenta de que es necesario reparar en ciertos puntos que a priori no tenían tanta importancia:

**Seguridad de los datos:** Al trabajar con datos de carácter privado (DNI's, NIF's, etc...) es importante mantener dicha privacidad, ampliando la seguridad del sistema, para ello algunas soluciones pueden ser:

- Seguridad a nivel de red: Uso de firewall a nivel de red, permitiendo solo el acceso al puerto 80 (HTTP) por los usuarios, y , opcionalmente, dando permiso a alguna ip de un administrador a los puertos usados por MySQL o alguna otra utilidad (Escritorio remoto, etc).
- A nivel de aplicación, es necesario reestructurar la aplicación para que la seguridad no solo sea a nivel de aplicación, sino que los permisos de los usuarios afecten directamente a las consultas, no permitiendo que ningún objeto que llegue al usuario contenga información para la que no tiene permiso.

**Integridad de los datos:** Es necesario reforzar cosas como copias de seguridad, solidez de los datos guardados, etc... Ya que al ser usados para la gestión, un simple error de datos puede ser muy problemático para el usuario.

Si nos paramos a pensar en un sistema global para la investigación, las ampliaciones anteriormente descritas no son suficientes, si hacemos un estudio un poco más exhaustivo sobre la evolución del sistema, sería, de forma resumida, algo así:

*¿Qué hemos hecho hasta ahora?*

- Base de datos relacional que comprende la información de proyectos, organismos, grupos de investigación e investigadores.
- Modulos que dan servicios básicos y algunos avanzados sobre la información anterior.
- Web que facilita el acceso a los servicios provistos.

***¿Qué habría que ampliar para conseguir un sistema global de la investigación?***

- Ampliar la base de datos, para reunir la información que no se tenía en cuenta (memorias de investigación, etc.)
- Como dijimos antes, hacer un verdadero estudio de la seguridad telemática del sistema.
- Ampliar los modulos de servicios, pensando en nuevos servicios que pueden darse al usuario
- Ampliar los conceptos de plataformas:
  - Nueva plataforma: Observatorio de investigación, orientado a cualquier usuario.
  - Ampliación de la gestión: ampliar la gestión a todos los tipos de datos, y crear diferentes tipos de gestión de forma segura, por ejemplo, crear perfiles de usuarios administradores de personal, otros de proyectos, etc.
  - Ampliación de los análisis, los nuevos análisis pueden ser aún mucho más potentes que los usados en ACUSE.
- Pensar en los nuevos recursos que se van a necesitar, (Servidor más potente, herramientas de mantenimiento, como, por ejemplo, para copias de seguridad de los datos, etc.)

### A1.1 Introducción

En esta guía se pretende explicar de forma resumida el uso y utilidad del Siscue, como se ha podido ver a lo largo de este documento Siscue es un sistema de información orientado a la investigación en la UPM.

El resumen de la estructura de nuestro sistema es el siguiente:

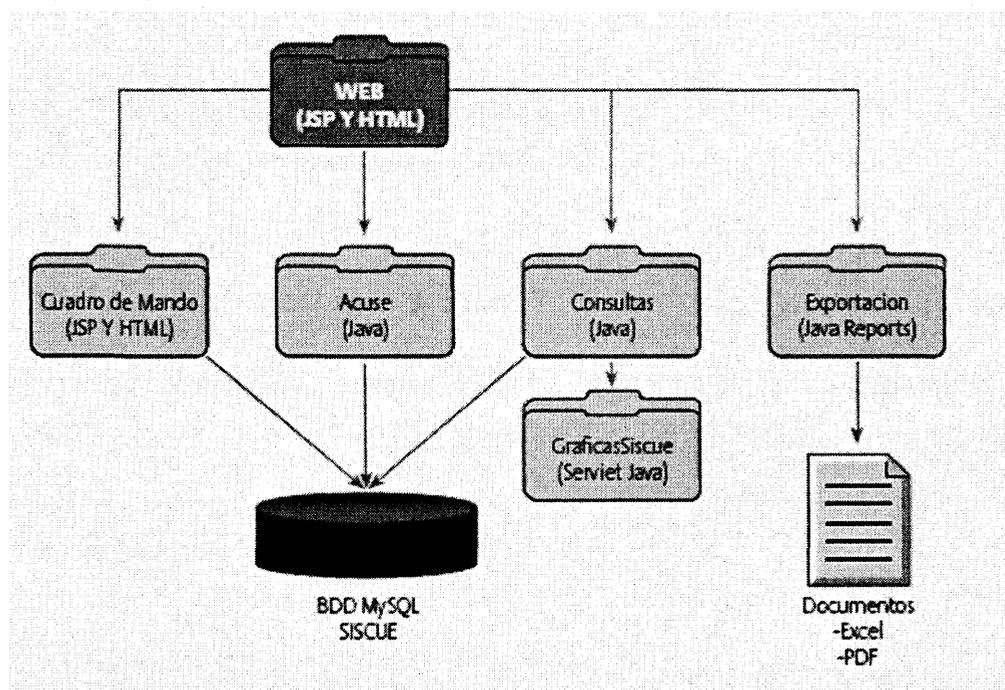


FIGURA A1-1: RESUMEN DEL SISTEMA

Donde las tareas que podemos realizar se resumen en los siguientes bloques:

- Navegación básica por los datos
- Administración de los datos (Modificado, agregado y eliminado)
- Navegación avanzada (búsquedas complejas, consultas estadísticas, etc.)
- Exportación de datos (a formatos como Pdf de Acrobat, o Excel de Microsoft Office)

## **A1.2 Instalación del sistema**

Para poder poner en funcionamiento el Siscue necesitaremos lo siguiente:

- PC servido donde se instale la aplicación para dar servicio, con conexión a la red donde se quiera hacer uso del sistema.
- Software de servidor web.
- Software de servidor MySQL.
- JRE de java (versión 1.4 o superior).
- Script de la base de datos acuse: **acuse.sql**.
- Plataforma Siscue comprimida para desplegar en servidor web (**siscue.war**).

En esta instalación vamos a usar JRE 1.5, Apache Tomcat 5.0 de servidor web, y MySQL 5.0 server para la base de datos.

### **A1.2.1 Instalación del servidor MySQL5 y la base de datos**

Podemos bajarnos la versión gratuita de MySQL5 de la web oficial ([www.mysql.com](http://www.mysql.com)). Una vez instalado en nuestro equipo debemos subir la base de datos al servidor MySQL, para ello podemos hacerlo de varias formas, algunas de ellas pueden ser:

#### **Forma 1: Con un cliente MySQL sencillo.**

Podemos descargarnos también de la web oficial de MySQL un administrador cliente para el servidor, llamado MySQL administrator. Una vez instalado lo ejecutamos, apareciendo una pantalla para hacer una conexión al servidor de la base de datos (Que hemos instalado con anterioridad).

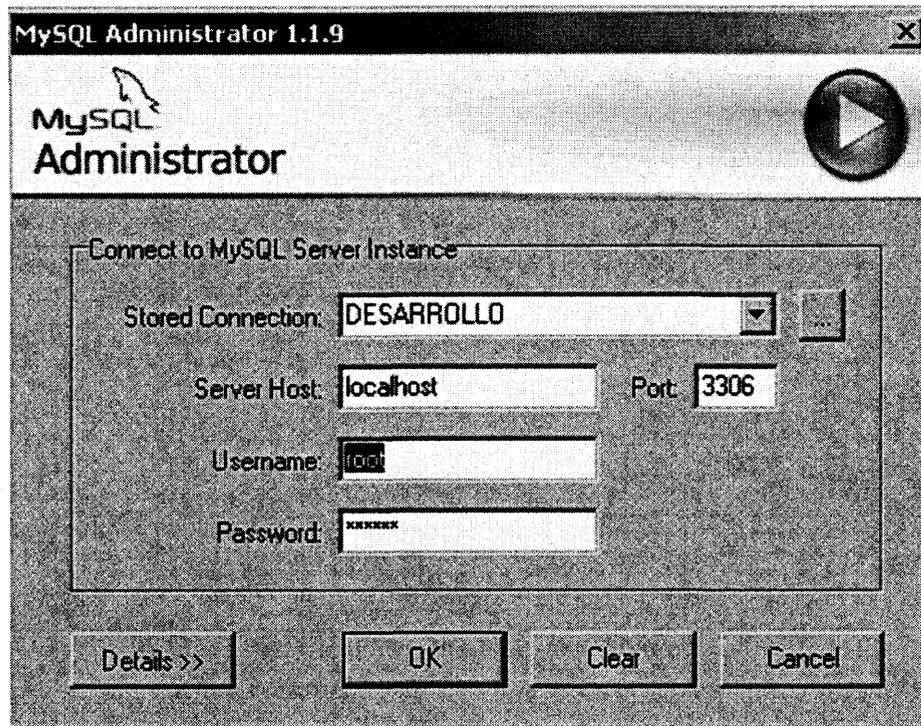


FIGURA A1-2: MYSQL ADMINISTRATOR, ACCESO

Los datos que necesitamos poner ahí son:

- **Stored Connection:** Nos permite guardar una configuración de conexión habilitando un nombre.
- **Server Host:** Dirección ip de donde está instalado el servidor MySQL, si es el mismo pc desde el que estamos conectando, nos vale con poner localhost, si es otro pc, debemos facilitar su dirección ip y tener acceso a dicha dirección.
- **Port:** Puerto donde escucha el servidor MySQL, por defecto MySQL usa 3306.
- **Usuario:** La primera vez debemos usar el usuario root creado en la instalación del servidor MySQL, una vez accedamos al servidor MySQL, podremos crear diferentes tipos de usuarios.
- **Password:** Contraseña para el usuario facilitado.

Una vez conectados al servidor, tenemos que tener a mano el script de la base de datos acuse (acuse.sql), para poder importarlo a nuestro servidor.

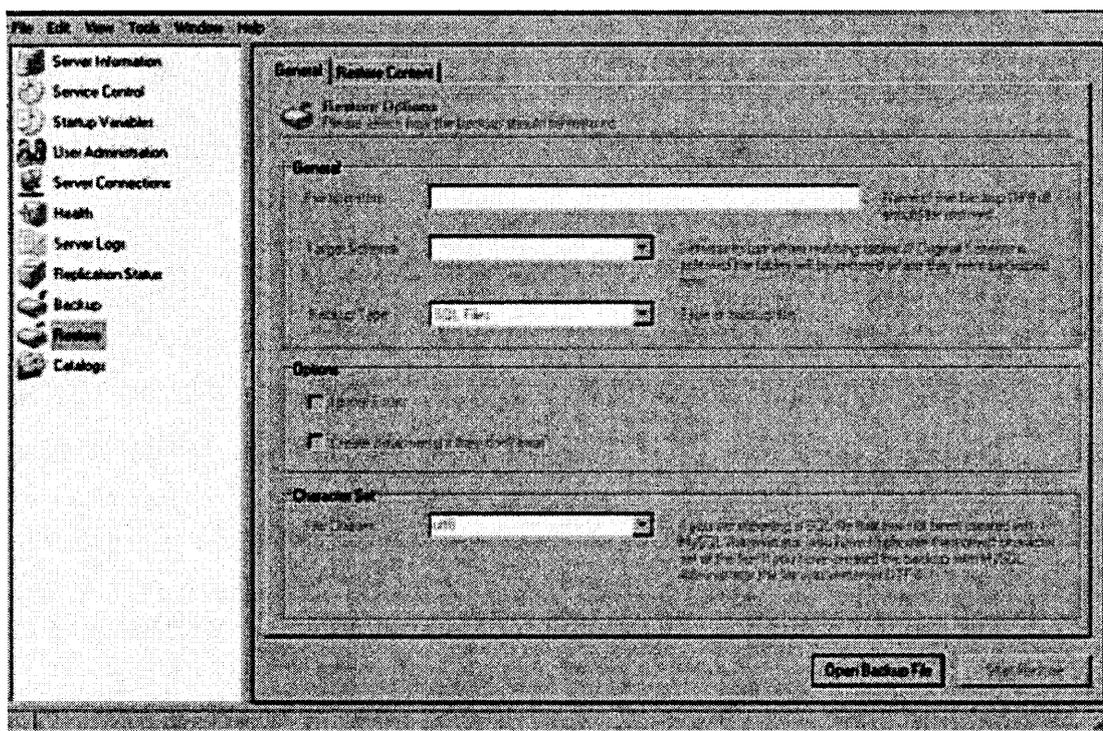


FIGURA A1-3 MYSQL ADMINISTRATOR, OPEN BACKUP FILE

Elegimos la opción “Restore”, y una vez dentro la opción “Open Backup File” donde podremos buscar el Script de la base de datos que vamos a importar, en este caso buscamos `acuse.sql` donde lo tengamos guardado.

Una vez seleccionado el archivo le damos a “Start restore” y esperamos a que la importación halla finalizado, y entonces ya tendremos la base de datos lista.

Hay que crear un usuario que es el que usa la aplicación Siscue por defecto, con privilegios sobre la base de datos `acuse` que hemos importado anteriormente, dicho usuario es:

- **Nombre:** `acuse`
- **Password:** `acuse`

**Forma 2: por consola**

Donde tengamos instalado el MySQL Server 5 (por defecto C:\Archivos de programa\MySQL\MySQL Server 5.0 ) hay una carpeta con los ejecutables llamada bin, así que por consola podremos ejecutar lo siguiente:

```
mysql -u user -p acuse < acuse.sql
```

nos pedirá el password del usuario que pongamos en la palabra "user" si hemos creado el usuario acuse con password acuse, ponemos en user: acuse, y el password acuse cuando nos lo pidan.

### **A1.2.2 Instalación del Apache Tomcat 5.0 y despliegue de la plataforma Siscue**

Apache Tomcat es un servidor web gratuito, que podemos encontrar en <http://tomcat.apache.org/> .Hay numerosas versiones de tomcat, con diferentes aplicaciones e interfaces para el usuario, es recomendable ver la documentación del Tomcat que se descarge.

Una vez descargado, instalado y ejecutado el Tomcat, podemos acceder a la página inicial de la siguiente forma:

<http://localhost:8080> , esta url es si tenemos instalado el Tomcat en la misma máquina desde la que estamos accediendo, sino deberíamos cambiar localhost por la ip de la máquina.

El Tomcat pone las aplicaciones por defecto en el puerto 8080, si se quiere cambiar al puerto universal web (para no tener que escribirlo en una URL) se puede hacer desde el archivo server.xml de la carpeta config del mismo (Ver documentación de Tomcat para más información).

Al acceder a la url anterior muestra lo siguiente:

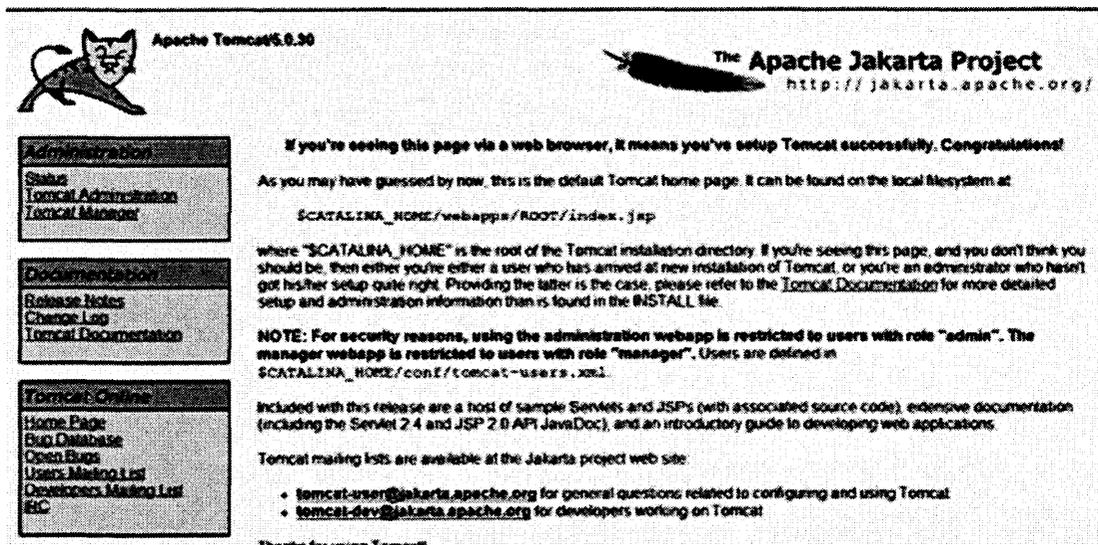


FIGURA A1-4: TOMCAT INICIO

Donde la aplicación que nos va a servir para desplegar la aplicación del Siscue será el Tomcat Manager. Para acceder al manager nos pedirá usuario y contraseña, donde el usuario por defecto es "admin" y el password deberíamos haberlo facilitado en la instalación del Tomcat. Dentro del manager usaremos la opción desplegar para cargar la aplicación del Siscue:

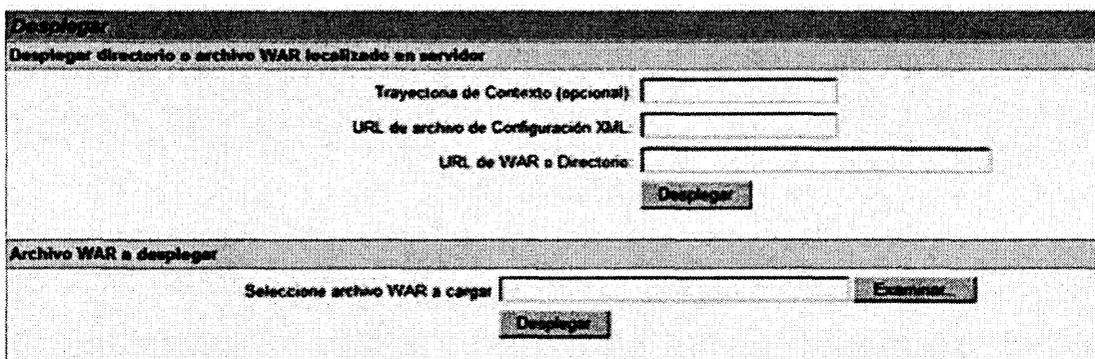


FIGURA A1-5: TOMCAT MANAGER

En la opción "Archivo War a desplegar" usamos el botón examinar para buscar el archivo Siscue.war por nuestro equipo, una vez seleccionado damos a desplegar. Si la configuración del tomcat es la inicial la url para acceder al Siscue será:

<http://ip:8080/Siscue><sup>3</sup>

donde ip es la dirección ip de la máquina que tiene instalada el servidor Tomcat.

### **A1.3 Navegación básica por los datos**

Para entrar al sistema nos encontramos con un panel de acceso (nombre de usuario y contraseña) del que deberemos disponer una cuenta activa para poder acceder.



FIGURA A1-6: ACCESO AL SISTEMA

Una vez dentro tenemos dos elementos que van a ser comunes en cada todo el sistema y que nos ayudarán en una navegación más rápida e intuitiva, dichos elementos son la barra lateral y superior.

#### **A1.3.1 Barra Superior**

---

<sup>3</sup> **NOTA:** Es recomendable activar SSL en el servidor web y obligar a la aplicación del Siscue a ser solo accesible por SSL, para más información, ver documentación de configuración SSL en Tomcat.



FIGURA A1-7: NAVEGACION, BARRA SUPERIOR

Consta de un menú desplegable, las opciones son:

- **Proyectos**
  - Todos
  - Artículo 83
  - Europeos
  - Nacionales
  - Internacionales
  - Regionales
  - Título propio
  - Cátedras
  - Licencias
  - Convenios
  
- **Organismos**
  - Todos
  - Universidades
  - Empresas
  - Entidades I+D+i
  - Administraciones
  - Otros
  
- **Investigadores**
  - Todos
  - Principales
  
- **Estadísticas**
  
- **Grupos de Investigación**
  - Grupos
  - Actividades de investigación

### A1.3.2 Barra Lateral

**BUSCADORES**

**Proyectos:**  
 » 

**Organismos:**  
 »

**Invs. en grupos de inv.:**  
 »

**Investigadores:**  
 »

**Grupos Inv :**  
 » 

**Actividades Grupos:**  
 »

para una búsqueda avanzada de elementos pinchar en el más.

---

**Miguel Angel Lucas**  
[Perfil](#)

**Cuadros de Mando:**

**Europeos:**  
[Cuadro FP6](#)

**Grupos de inv.:**  
[Plan Calidad](#)

---

[Inicio](#)  
[Desconectar](#)

La barra lateral dispone de buscadores para todos los elementos principales (proyectos, organismos, investigadores, grupos de investigación y actividades). Al usarlos directamente nos lleva a una lista de los elementos correspondientes pero buscados por las palabras puestas en la casilla de su buscador.

Además aparece un  a la derecha de los elementos que disponen de una "búsqueda avanzada", donde no solo se buscará por palabra, sino que se configurará un panel entero para sacar consultas mucho más complejas.

Esta barra también nos proporciona acceso a los cuadros de mando de diferentes categorías (como el Plan de Calidad o un cuadro orientado al Sexto Programa Marco). Así mismo podemos acceder a los elementos de administración de agregado de todos los elementos (agregar un proyecto, organismo, etc.)

Por último nos aparece los datos del usuario que ha accedido al sistema y un link a su perfil, donde podrá modificar los datos personales y ver los permisos de los que dispone.

Haciendo uso de estas dos barras y de los elementos que nos vayan apareciendo en el sistema, podemos navegar hacia todas las direcciones posibles.

### A1.3.3 Elementos:

#### → Proyectos

Podemos acceder a ellos desde:

- Listados de proyectos.
- Barras de navegación.
- Elementos relacionados:
  - Investigadores que participen en proyectos.
  - Organismos que formen parte del consorcio de proyectos.
  - Investigadores pertenecientes a un grupo de investigación.
- Consultas estadísticas sobre proyectos

En los listados de proyectos podemos crear una lista de los elementos por filtros, por ejemplo, podemos seleccionar los proyectos Artículos 83 del 2006 de un centro específico (además de esto siempre se puede buscar por palabra clave desde la barra lateral):

The screenshot shows a search interface for projects. At the top right, it indicates 'Página: 1' and 'Hay 39 Proyectos'. Below this, there are four filter fields: 'Marco de Colaboración' with a dropdown menu showing 'Artículo 83', 'Centro' with a dropdown menu showing 'ETSI DE MINAS', 'Años' with a dropdown menu showing '2006', and 'Departamento' with a dropdown menu showing 'Todos'. Below the filters is a table header with three columns: 'Código', 'Título', and 'Marco'.

FIGURA A1-8: PROYECTOS

#### Filtros:

- Marco de colaboración: Tipo de proyecto, como Europeo.
- Submarco: Dentro del tipo de proyecto, como FP6 dentro de Europeo.
- Año: Año de concesión del proyecto.
- Centro: Centro UPM al que pertenece el proyecto.
- Departamento: Departamento UPM al que pertenece el proyecto.

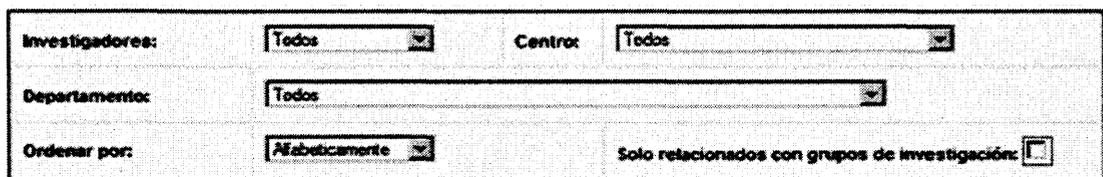
Desde las barras accedemos directamente a listados con una configuración fija o por búsqueda de palabras clave. Desde las listas o todo elemento relacionado en el que exista un link a un proyecto podemos ir directamente a la ficha resumen del proyecto en cuestión.

### → Investigadores

Parecido a proyectos podemos acceder a estos elementos desde:

- Listados de investigadores.
- Barras de navegación.
- Elementos relacionados:
  - Proyectos en que participen investigadores.
  - Grupos de investigación que tengan investigadores relacionados
- Consultas estadísticas sobre investigadores

Los filtros que tenemos para investigadores son (además de la búsqueda por palabra clave de la barra lateral):



The image shows a filter interface for investigators. It contains four main sections: 'Investigadores:' with a dropdown menu set to 'Todos'; 'Centro:' with a dropdown menu set to 'Todos'; 'Departamento:' with a dropdown menu set to 'Todos'; and 'Ordenar por:' with a dropdown menu set to 'Alfabeticamente'. To the right of the 'Ordenar por:' section is a checkbox labeled 'Solo relacionados con grupos de investigación:' which is currently unchecked.

FIGURA A1-9: FILTROS DE PROYECTOS

### Filtros:

- Tipo de investigadores: Todos o responsables.
- Solo relacionados con grupos: Solo los que pertenezca a algún grupo de investigación.
- Centro: Centro UPM al que pertenece el investigador.
- Departamento: Departamento UPM al que pertenece el investigador.

### → Organismos

Podemos acceder a estos elementos desde:

- Listados de organismos.
- Barras de navegación.
- Elementos relacionados:
  - Proyectos con consorcio.
- Consultas estadísticas sobre organismos.

Los filtros que tenemos para organismos son (además de la búsqueda por palabra clave de la barra lateral):

Organismos:	<input type="text" value="Empresa"/>	Tipo:	<input type="text" value="Gran Empresa (&gt;250)"/>
Nacionalidad:	<input type="text" value="Estranjeros"/>	País:	<input type="text" value="Austria"/>

Nombre Organismo  País  Cif

FIGURA A1-10: ORGANISMOS

### Filtros:

- Organismo: Tipo de organismo, como Empresa.
- Tipo: Subtipo, por ejemplo dentro de Empresa, elegir Pequeña Empresa.
- Nacionalidad: Española o extranjera.
- País: Si es extranjera se puede especificar su país.

### → Grupos de investigación

Podemos acceder a estos elementos desde:

- Listados de grupos de investigación.
- Barras de navegación.
- Elementos relacionados:
  - Investigadores que pertenecen a grupos.
- Consultas estadísticas sobre grupos de investigación.

Los filtros que tenemos para grupos son (además de la búsqueda por palabra clave de la barra lateral):

123 Página: 1 May 215

Por nombre:

Ordenar Por:

Lineas:  Areas:  Estado:

FIGURA A1-11: FILTROS ORGANISMOS

**Filtros:**

A diferencia de los otros elementos, en grupos de investigación podemos buscar otros tipos de palabras clave, los relacionados con las Areas temáticas y las líneas de investigación

- *Lineas:* Palabra clave de líneas de investigación que tenga el grupo.
- *Areas:* Areas temáticas relacionadas con el grupo.
- *Estado:* Consolidado, en proceso, de baja o desconocido.

Todos los elementos tienen una ficha resumen de dicho elemento, accesible desde cualquier parte donde aparezca (con un link).

**A1.3.4 Estadísticas**

Además de esta navegación básica podemos navegar por el sistema para realizar estadísticas o análisis. Una de las partes que nos permite hacer esto es accesible desde el botón "Estadísticas" de la barra superior.

Si pulsamos en Estadísticas nos aparece un panel sencillo como el siguiente:

Consulta en torno:

Tipo de Proyectos:  Tablas

Año/s Proyectos:  Graficas

Presupuesto:

FIGURA A1-12: CONFIGURACIÓN DE CONSULTAS SENCILLAS (I)

Estas estadísticas están siempre orientadas para consultas sobre proyectos, así que podemos definir sobre que lo queremos:

- Tipo de proyectos (Ej: Europeo).
- Subtipo de proyectos (Ej: FP6).
- Año/s del proyecto: Los de acuse. 2003-2006, o cada uno por separado.
- Presupuesto: Nos da las siguientes opciones:
  - <6000.
  - 6000-60000.
  - 60000-600000.
  - >600000.

Y por ultimo si queremos sacar una tabla detallada o una gráfica de lo seleccionado.

Para cada elemento en torno al que queramos hacer la consulta, se despliega un panel específico, por ejemplo para Centros UPM es el siguiente:

**HA SELECCIONADO CENTROS**

Tipo de Consulta:

Ordenar por:

**Se mostrará el resultado en Gráficas**

Tipo de gráficas:

Número de Valores\*

\* Si el número supera el máximo de la consulta, sacará dicho máximo, en blanco coge por defecto 10 valores

FIGURA A1-12: CONFIGURACIÓN DE CONSULTAS SENCILLAS (II)

Una vez configurado esto, al dar al botón, el sistema nos genera una consulta con los parámetros elegidos.

Ejemplo de consulta (Datos ficticios):

Parámetros:

**Consulta en torno a** → CENTROS

**Tipo de proyectos** → Europeo.

**Subtipo de proyectos** → FP6.

**Año** → 2006.

**Presupuesto** → Todos.

**Tipo de consulta** → Número de proyectos

En gráfica de Queso de 10 valores

Resultado:

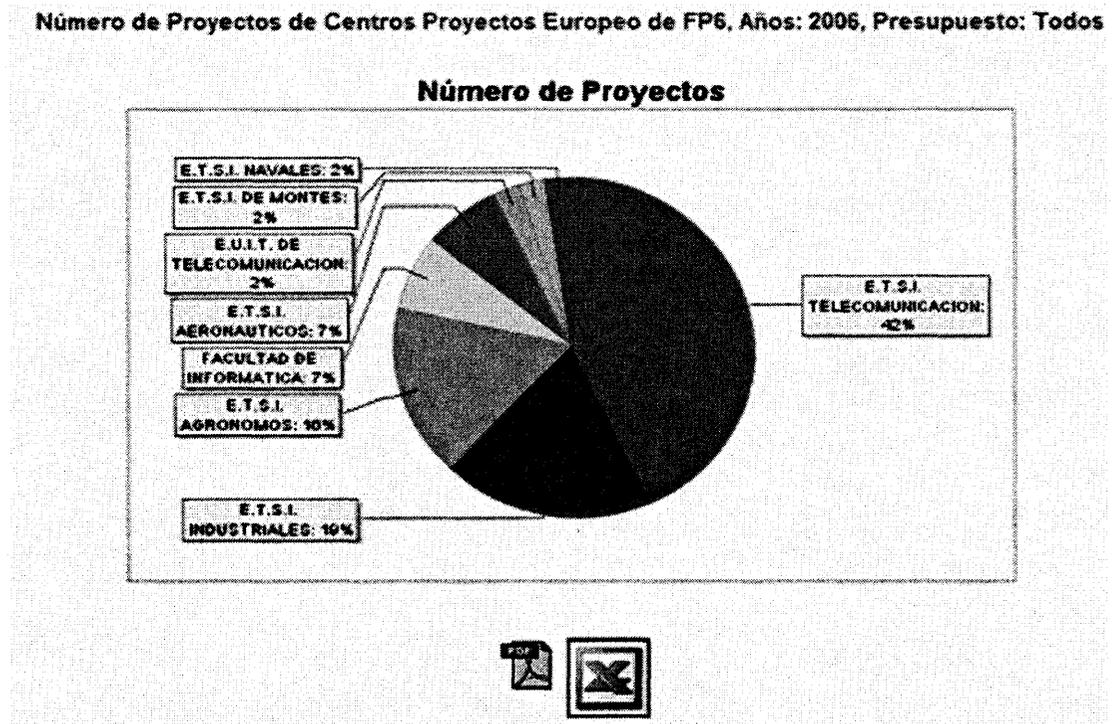


FIGURA A1-13: EJEMPLO DE RESULTADO DE CONSULTAS SENCILLAS

### **A1.3.5 Busquedas avanzadas de elementos**

En fecha de escritura de este documento existían dos tipos de búsquedas avanzadas:

De proyectos

De grupos de investigación.

Ambas siguen la misma lógica para para diferentes elementos, permitiendo:

Elegir valores que se necesitan

Filtrar por las características del elemento en cuestión.

Filtrar y mostrar las relaciones con otros elementos

Una vez hecha la búsqueda avanzada nos deja también exportar los datos a Excel.

#### **A1.3.5.1 Búsqueda avanzada de proyectos**

El panel para este objetivo (accesible desde el + a la derecha del buscador de proyectos) es el siguiente:

### Búsqueda Avanzada de Proyectos

**Palabra Clave:**

**Marco de colaboración:**

**Marco específico:**

**Años Proyectos:**

**Presupuesto:**



**POLITÉCNICA**

**Datos a mostrar del proyecto:**

<input type="checkbox"/> Título	<input type="checkbox"/> Tipo de Proyecto (I+D, formación,...)
<input type="checkbox"/> Código OTT	<input type="checkbox"/> Centro
<input type="checkbox"/> Investigador Principal	<input type="checkbox"/> Departamento
<input type="checkbox"/> Marco de Colaboración (Europeo,Nacional,Art 83...)	<input type="checkbox"/> Presupuesto concedido UPM
<input type="checkbox"/> Submarco (Europeo: FP6,EUREKA...)	<input type="checkbox"/> Programa
<input type="checkbox"/> Año	<input type="checkbox"/> Instrumento
<input type="checkbox"/> Observaciones	<input type="checkbox"/> Área de conocimiento

FIGURA A1-14: CONFIGURACION DE BUSQUEDA AVANZADA DE PROYECTOS (I)

**Buscar sólo los proyectos que tengan relación al menos con los siguientes organismos:**

Tipo:

Nacionalidad:

→ País:

**Datos a mostrar del organismo:**

<input type="checkbox"/> Nombre Organismo	<input type="checkbox"/> CIF
<input type="checkbox"/> Tipo de organismo (General)	<input type="checkbox"/> País
<input type="checkbox"/> Tipo de organismo (Específico)	<input type="checkbox"/> Datos de contacto (Email, teléfono, dirección,...)

**NOTA:** Si selecciona algún campo a mostrar de organismos relacionados, el sistema buscará todas las relaciones con los organismos deseados (existan o no) y mostrará los campos seleccionados, pudiéndose ver proyectos repetidos para mostrar las diferentes relaciones con todos los organismos asociados.

**Buscar sólo los proyectos que pertenecen a:**

Centros:

**NOTA:** Debido a la posibilidad de crear consultas de tamaño considerable, no se pueden insertar links de los elementos a sus fichas.

Exportar directamente a Excel

FIGURA A1-15: CONFIGURACION DE BUSQUEDA AVANZADA DE PROYECTOS (II)

Como se puede observar en la imagen podemos seleccionar sobre los proyectos:

**Palabra clave:** una palabra en el título del proyecto

**Marco de colaboración:** Tipo de proyecto (Europeo, Nacional, etc)

**Marco específico:** FP6, Eureka, etc.

**Años :** Todos o alguno en específico

**Presupuesto:** los mismos que en estadísticas.

Una vez elegidos los parámetros deseados para los proyectos, podemos seleccionar que queremos que nos saque de ellos (se ven en "Datos a mostrar del proyecto")

Así mismo podemos elegir proyectos relacionados con un tipo de organismo específico (Empresa, Entidad I+D+i, etc.) y los datos del organismo relacionado.

Y por último ver solo los proyectos relacionados con un centro y/o departamento UPM en específico.

Ejemplo de una búsqueda avanzada de proyectos (datos ficticios):

- **Palabra clave:** tec
- **Marco de colaboración:** Todos
- **Años :** Todos
- **Presupuesto:** Todos

Sobre organismos:

- **Tipo:** Empresa
- **Tipo específico:** Todos
- **Nacionalidad:** Extranjera
- **Pais:** Todos Extranjero

- **Centro:** Todos

Y que nos muestre:

de los proyectos:

- Título
- Investigador Principal
- Marco de colaboración
- Submarco
- Tipo de proyecto

de los organismos:

- Nombre organismo
- Datos de contacto
- País

Resultado (cortado para poder mostrarlo):

Orden	Título	Tipo	Investigador/Responsable	AmbitoGeografico	MarcoColaboracion	NombreOrganismo	PaisOrganismo
1	Threat Hunting of Dark Spikes - 2	Proyecto	...	Europe	...	...	República Checa
2	Threat Hunting of Dark Spikes - 1	Proyecto	...	Europe	...	...	Francia

**FIGURA A1-16: RESULTADO DE BUSQUEDA AVANZADA DE PROYECTOS**

### A1.3.5.2 Búsqueda de grupos de investigación

Al igual que para proyectos, para grupos se accede a este panel desde el + a la derecha del buscador de grupos, mostrando lo siguiente:

**Búsqueda Avanzada de Grupos de Investigación**

Palabra Clave:

Estado:

Convocatoria:

**Datos a mostrar del Grupo de Investigación:**

<input type="checkbox"/> Nombre	<input type="checkbox"/> Observaciones
<input type="checkbox"/> Investigador Responsable	<input type="checkbox"/> Centro IR
<input type="checkbox"/> Departamento IR	<input type="checkbox"/> Código OTT
<input type="checkbox"/> Otros Centros Implicados	<input type="checkbox"/> E-mail IR
<input type="checkbox"/> Acrónimo	<input type="checkbox"/> E-mail Grupo
<input type="checkbox"/> Convocatoria	<input type="checkbox"/> Teléfono
<input type="checkbox"/> Estado	<input type="checkbox"/> Fax
<input type="checkbox"/> Web	<input type="checkbox"/> Número de registro
<input type="checkbox"/> Fecha de registro	

**POLITÉCNICA**

FIGURA A1-17: CONFIGURACION DE BUSQUEDA AVANZADA DE GRUPOS (I)

NOTA: Si marca la opción de Áreas o Líneas, buscará las relaciones por la palabra clave insertada (Todas las relaciones si la palabra clave está en blanco) mostrando grupos repetidos para cada relación.

Áreas de Investigación  
 Palabra clave:

Líneas de Investigación  
 Palabra clave:

**Ayudas:**

Mostrar solo grupos que tienen Ayudas

Convocatoria:  Recuerde que si elige alguna categoría en especial de las ayudas a grupos, el Sistema sacará solo los grupos que tienen ayudas en esa convocatoria.

**Datos a mostrar de las Ayudas:**

<input type="checkbox"/> Título ayuda	<input type="checkbox"/> Referencia
<input type="checkbox"/> Cantidad concedida	<input type="checkbox"/> Concedido gastos ejecución
<input type="checkbox"/> Concedido becas	<input type="checkbox"/> Total importe seleccionado
<input type="checkbox"/> Convocatoria de la Ayuda	

NOTA: Si selecciona algún campo a mostrar de ayudas relacionadas, el sistema buscará todas las relaciones con las ayudas deseadas (existan o no) y mostrará los campos seleccionados, pudiéndose ver grupos repetidos para mostrar las diferentes relaciones con todas las ayudas asociadas.

FIGURA A1-18: CONFIGURACION DE BUSQUEDA AVANZADA DE GRUPOS (II)

**Personal del grupo de investigación:**

Categorías:

Doctor:

**Datos a mostrar del personal:**

<input type="checkbox"/> Nombre	<input type="checkbox"/> Apellidos
<input type="checkbox"/> Categoría	<input type="checkbox"/> Email
<input type="checkbox"/> Teléfono	<input type="checkbox"/> Centro
<input type="checkbox"/> Departamento	<input type="checkbox"/> DNI
<input type="checkbox"/> Ex doctor	

NOTA: Si selecciona algún campo a mostrar de personal relacionado, el sistema buscará todas las relaciones y mostrará los campos seleccionados, pudiéndose ver grupos repetidos para mostrar las diferentes relaciones con todo su personal.

**Buscar sólo los grupos que pertenecen a:**

Centros:

NOTA: Debido a la posibilidad de crear consultas de tamaño considerable, no se pueden insertar links de los elementos a sus fichas.

Exportar directamente a Excel

FIGURA A1-19: CONFIGURACION DE BUSQUEDA AVANZADA DE GRUPOS (III)

Aquí podemos configurar:

**Datos de los grupos:**

- Palabra clave
- Estado
- Convocatoria

**Otros datos:**

- Áreas temáticas y líneas de investigación: Se puede seleccionar que nos saque también las áreas y/o líneas relacionadas con el grupo de investigación, incluso buscando por palabras.
- Ayudas a grupos: Sobre todo para la gestión, ayudas económicas asociadas a grupos.
- Personal implicado: Se puede filtrar personal de los grupos por:
  - Categoría del investigador
  - Doctor

De todos los elementos podemos seleccionar que queremos que nos saque en el listado.

Ejemplo de una búsqueda avanzada de grupos de investigación (datos ficticios):

- **Palabra clave:**
- **Estado:** Consolidado
- **Convocatoria:** Todas

No marcamos áreas temáticas.

Marcamos líneas de investigación.

- **Palabra Clave (De líneas):** Tele

No marcamos ver responsables de las líneas.

No marcamos ayudas.

- **Categoría:** PROF. CATEDRÁTICO UNIV
- **Centro:** Todos

Y que nos muestre:

de los grupos:

- Nombre del grupo
- Investigador Responsable
- Centro IR

del personal:

- Nombre
- Apellidos
- Categoría
- Centro

Resultado (cortado para poder mostrarlo):

Order	Link	Nombre_Grupo_Investigacion	Nombre_Investigador_Principal	Apellido_Investigador_Principal	ControlR	Linea
1	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
2	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
3	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
4	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
5	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
6	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
7	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
8	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
9	<a href="#">Bcha</a>	Biogenoma Aplicada	CARLOS	PLATERO DUEÑAS	E.I.I.T. INDUSTRIAL	Telemedicina
10	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación
11	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación
12	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación
13	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación
14	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación
15	<a href="#">Bcha</a>	Centros de la Computación en la Educación y en la Representación del Conocimiento	JOSE ALBERTO	JAVEN GALLEGO	E.T.S.I. INDUSTRIALES	Teleeducación

FIGURA A1-20: RESULTADO DE BUSQUEDA AVANZADA DE GRUPOS

### A1.3.6 Plan de calidad

Para acceder al Plan de calidad basta con pulsar el botón de Plan de Calidad en los cuadros de mando de la barra lateral. Al pulsarlo se nos abrirá la siguiente página:

SEÑALES	2006.0	1000.0
Formación de investigadores (p. inv)	0.25	0.25
Difusión de resultados (p. inv)	0.6	0.3
Explotación de resultados (p. grupo)	1.0	1.0

Resumen de grupos y umbrales	
No pasan ningún	9
Pasan un umbral	29
Pasan dos umbrales	51
Pasan tres umbrales	100

FIGURA A1-21: CONFIGURACIÓN DEL PLAN DE CALIDAD

Configuración del plan de calidad:

- **Año:** Año sobre el que queremos hacer la calidad de la investigación
- **Ver:** Datos que queremos ver, opciones:
  - *Resultados Finales:* Muestra solo el porcentaje y peso total de los grupos de investigación
  - *Solo grupo de indicadores:* Muestra los resultados finales y los grupos de indicadores (Recursos económicos, formación de personal investigador, difusión de resultados, explotación y reconocimiento de méritos).

- *Todo*: Muestra todo lo anterior y además los indicadores de cada grupo, por ejemplo en formación de personal investigador las tesis, becarios contratados, etc..
- **Financiación**: Aquí podemos configurar como va a ser la financiación total sobre los grupos configurados.
  - *Financiación total*: Total de presupuesto a repartir.
  - *Max. Por grupo*: Máximo que se puede adjudicar a un grupo.
  - *Min. Por grupo*: Mínimo que se puede adjudicar a un grupo.
- **UMBRALES**: En este panel podemos marcar los umbrales para ver que cantidad de grupos los pasan, separando los umbrales para grupos consolidados y en proceso de consolidación, si marcamos, por ejemplo, en generación de recursos económicos 2000 por investigador, nos dirá que grupos han conseguido pasar ese umbral y cuales no.
- **Mostrar calidad**:
  - *Por grupo*: La calidad la calcula por la actividad de todo el grupo, sin tener en cuenta el número de investigadores de cada grupo
  - *Por investigador*: La calidad la calcula por investigador, teniendo en cuenta el número de investigadores de cada grupo

Una vez configurado todo el panel, el sistema nos calcula el plan de calidad con los parámetros elegidos, mostrándonos una tabla navegable de los grupos y sus resultados.

### A1.3.7 Cuadro de mando Europeo FP6

El cuadro de mando Europeo FP6 es un resumen del Sexto programa marco de la información metida en el sistema, Para abrirlo basta con pulsar en el botón de los cuadros de mando de la barra lateral y nos aparece:

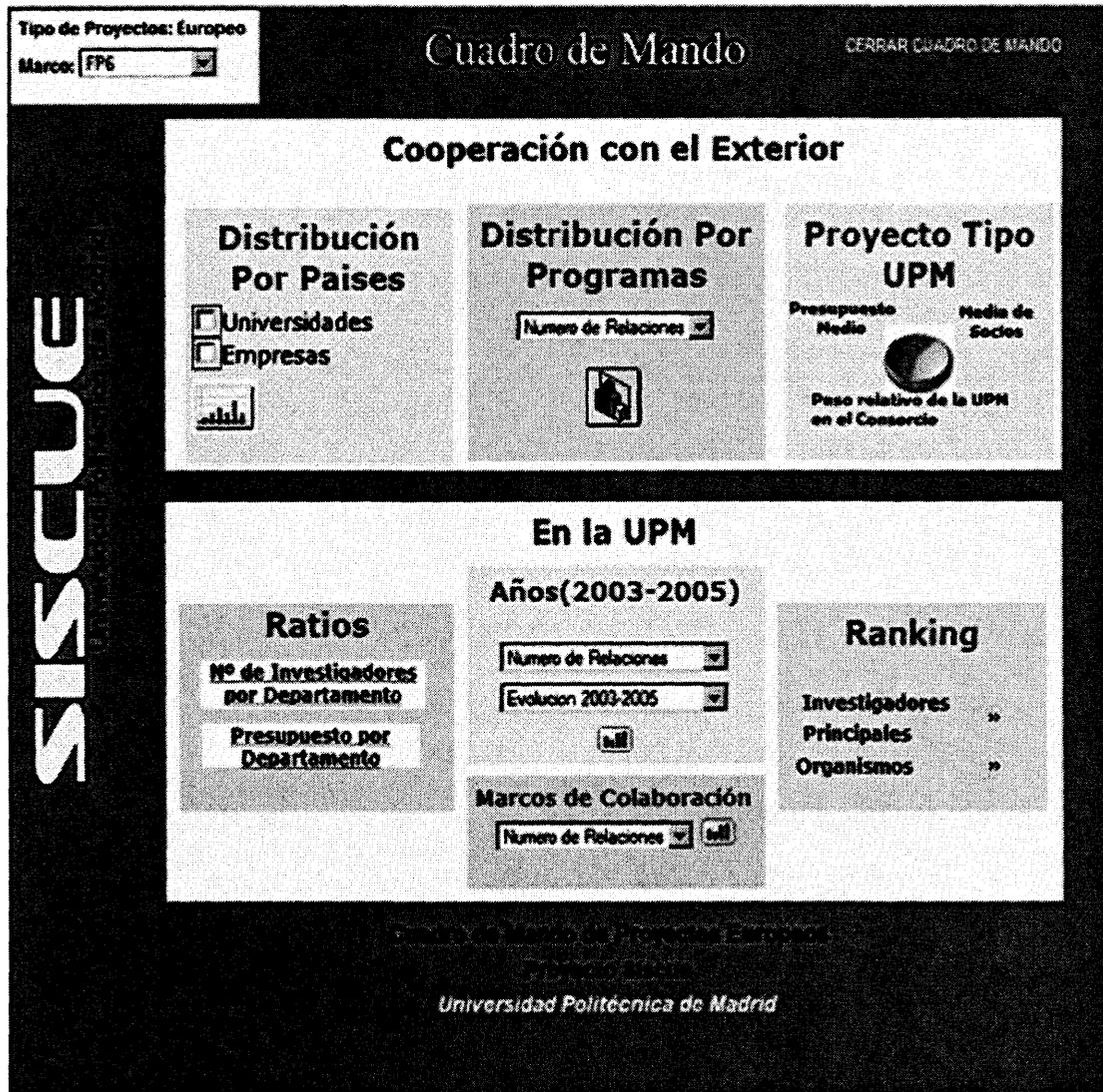


FIGURA A1-22: CUADRO DE MANDO EUROPEO

Desde este cuadro podemos ir navegando por las diferentes opciones para dar un "rápido vistazo" al FP6 en la UPM.

### A1.4 Administración de los datos

### A1.4.1 Modificación y borrado

Para la modificación y el borrado de un elemento basta con tener permisos de escritura sobre ellos, si es así, aparecerá en cada elemento la opción “modificar” o “borrar”, igualmente en las relaciones. El proceso de borrar puede no ser sencillo, pero para el usuario es transparente, de tal forma que si quiere borrar un investigador, se le avisará de que relaciones van a ser quitadas (por ejemplo, de relaciones con proyectos, grupos, etc). Ejemplo de modificado: **Proyecto**

Formulario de Proyecto	
Codigo del Proyecto	E-00-0000-00
Titulo del Proyecto	Proyecto de prueba
Año	2008
Fecha Inicio	6-3-2008
Fecha Entrada DTT	null
Rol de la UPM	Contractor
Observaciones	
Meses Duracion (NÚMERO ENTERO)	23
Ambito Geografico	Europeo
Marco de Colaboracion	FP7
Area	Vacio
Programa	IST
Instrumento	Coordinacion Action
Tipo de Proyecto	Proyecto I-D
Investigador Principal:	

FIGURA A1-23: MODIFICADO DE PROYECTO (I)

En esta ficha podemos modificar los datos relativos a un proyecto, y debajo de ella nos aparecen todos los datos relacionados con dicho proyecto:

Actualmente este proyecto pertenece a:  
Centro **E.U.I.T. DE TELECOMUNICACION**  
Departamento **MATEMATICA APLICADA A LA INGENIERIA TECNICA DE TELECOMUNICACION.**  
¿Desea cambiarlo?

Centro:

Modificar Investigador Principal

Asociar Investigador

Asociar Organismo

Nombre:	<input type="text"/>	Apellidos:	<input type="text"/>
Telefono:	<input type="text"/>	E-Mail:	<input type="text"/>
Cargo:	<input type="text"/>	Observaciones:	<input type="text"/>

EPO

Lista Investigadores Asociados

<input type="button" value="ficha"/>	GONZALO LEÓN SERRANO
<input type="button" value="ficha"/>	ANA GÓMEZ OLIVA

Lista Organismos Asociados

<input type="button" value="ficha"/>	CATSA - Centro de Asistencia Telefónica S.A.
--------------------------------------	--

FIGURA A1-24: MODIFICADO DE PROYECTO (II)

Donde podemos agregar cualquier elemento relacionado, borrar relaciones, etc.

#### A1.4.2 Agregado de elementos

Para agregar elementos tenemos que acceder desde el link de la barra lateral:

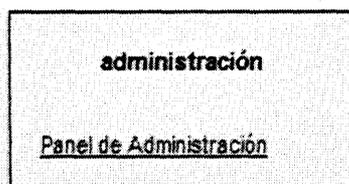


FIGURA A1-25: LINK ADMINISTRACIÓN

Que nos llevará a un pequeño panel donde nos mostrará las opciones en las cuales disponemos de permisos para poder llevarlas a cabo:

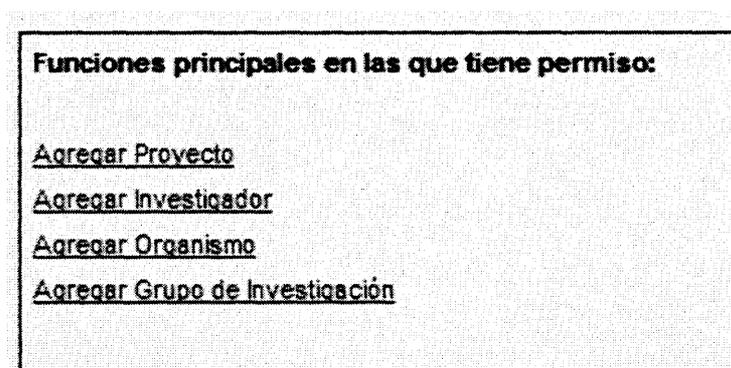


FIGURA A1-26: OPCIONES DE AGREGADO

Desde cada link accedemos a un formulario donde podemos agregar el elemento en cuestión.

Ejemplo de agregado de un proyecto:

Agregar Proyecto	
Título	<input type="text"/>
Cuenta OTT	<input type="text"/>
Año	2008 <input type="text"/>
Fecha Inicio	<input type="text"/> <input type="checkbox"/>
Fecha Entrada OTT	<input type="text"/> <input type="checkbox"/>
Papel Upm	<input type="text"/>
Meses Duración	<input type="text"/>
Observaciones	<input type="text"/>
Investigador Principal	<input type="text"/>
Ámbito Geográfico:	<input type="text" value="Elegir ámbito geográfico"/>
Marco de Colaboración:	<input type="text"/>
Programa:	<input type="text"/>
Instrumento:	<input type="text"/>
Centro:	<input type="text" value="NO ADSCRITOS A CENTRO"/>
Tipo de Proyecto	<input type="text" value="Sin tipo asignado"/>

FIGURA A1-27: AGREGADO DE PROYECTO (I)

The image shows a web interface for adding project elements. It consists of four main sections, each with a search field and an 'Asociar' button:

- Agregar Investigador Principal:** Search field and 'Asociar' button.
- Agregar Investigador Relacionado:** Search field and 'Asociar' button.
- Agregar Organismo Relacionado:** Search field and 'Asociar' button.
- Recursos Economicos:** Two input fields: 'Contribucion Máxima CE' and 'Presupuesto Concedido UPM'.

At the bottom of the form is a large 'Agregar' button.

FIGURA A1-23: AGREGADO DE PROYECTO (II)

Una vez relleno todos los datos y seleccionado todos los elementos que queremos relacionar, con dar al botón "Agregar" el sistema nos lo creará redireccionándonos a su ficha por si queremos modificar o completar alguna información.

## BIBLIOGRAFÍA Y REFERENCIAS

---

- Universidad Politécnica de Madrid – Vicerrectorado de Investigación:  
**Título:** “Análisis de la Cooperación de la Universidad Politécnica de Madrid con el Sector Empresarial”.**Autor:** Gonzalo León.
- Documentación de librerías POI: “The Apache Poi documentation” – URL en Enero del 2008: <http://poi.apache.org>
- Documentación de librerías JfreeChart: “JfreeChart API's” – URL en Enero del 2008: <http://www.jfree.org/jfreechart/>
- Libro sobre MySQL: **Título:** MySQL Avanzado. **Autor:** Derek J. Balling, Jeremy D. Zawodny. **Editorial:** Anaya Multimedia
- Libro sobre JSP: **Título:** JSP. **Autor:** Esteban Trigos García, **Editorial:** Anaya Multimedia.
- Libro sobre Java: **Título:** Java JDK 6. **Autores:** Donald Avondolio , W. Clay Richardson . **Editorial:** Anaya Multimedia.
- Libro sobre JavaScript: **Título:** JavaScript para desarrolladores web. **Autor:** Nicholas C. Zakas. **Editorial:** Anaya Multimedia.
- Portal de proyectos europeos CORDIS. Url en 2008: <http://www.cordis.com/>