



**Universitat Oberta  
de Catalunya**

---

# Memoria

---

TFC J2EE 2011-2012/2

Víctor Manuel Merino Martínez

# Índice de contenido

---

Introducción.....	6
Descripción del proyecto.....	6
Escenario.....	7
Esquema.....	7
Objetivos.....	8
Análisis de requisitos.....	9
Requerimientos.....	9
Módulo de socios/usuarios.....	9
Casos de uso.....	10
Inicio de sesión.....	10
Listar usuarios.....	10
Alta de usuario.....	11
Modificación de usuario.....	11
Informe de usuarios.....	11
Baja de usuario.....	12
Asignar grupo al usuario.....	12
Prototipos de interfaz gráfica.....	13
Inicio de sesión.....	13
Menú principal.....	13
Listado de usuarios.....	13
Alta de usuario.....	14
Modificación de usuario.....	15
Asignar grupo al usuario.....	16
Diagrama de flujo entre pantallas.....	17
Módulo de grupos.....	17
Casos de uso.....	18
Listar grupos.....	18
Alta de grupo.....	19
Modificación de grupo.....	19
Baja de grupo.....	19
Informe de grupo.....	20
Asignar usuarios al grupo.....	20
Prototipos de interfaz gráfica.....	20
Listado de grupos.....	20
Alta de grupo.....	21
Modificación de grupo.....	21
Asignar usuarios al grupo.....	22
Diagrama de flujo entre pantallas.....	23
Módulo de perfiles.....	23
Casos de uso.....	24
Listar perfiles.....	24
Alta de perfil.....	24
Modificación de perfil.....	25
Baja de perfil.....	25
Prototipos de interfaz gráfica.....	25
Listado de perfiles.....	25
Alta de perfil.....	26
Modificación de perfil.....	27

Diagrama de flujo entre pantallas.....	29
Módulo de componentes .....	29
Casos de uso.....	30
Listar grupos.....	30
Listar componentes.....	30
Modificación de componentes.....	31
Baja de componentes.....	31
Listar usuarios.....	31
Alta de componentes.....	32
Informe de componentes.....	32
Prototipos de interfaz gráfica.....	33
Listado de grupos.....	33
Listado de componentes.....	33
Listado de usuarios.....	33
Diagrama de flujo entre pantallas.....	34
Módulo de reuniones.....	34
Casos de uso.....	35
Listar reuniones.....	35
Enviar convocatoria.....	36
Generar acta.....	36
Eliminar reunión.....	36
Modificación de reunión.....	37
Prototipos de interfaz gráfica.....	37
Listado de reuniones.....	37
Alta/Modificación de reunión.....	37
Añadir datos/Modificación de reunión.....	38
Diagrama de flujo entre pantallas.....	40
Módulo de solicitudes.....	40
Casos de uso.....	41
Listar solicitudes.....	41
Editar/Revisar solicitud.....	41
Alta de solicitud.....	42
Comprobar estado solicitud.....	42
Prototipos de interfaz gráfica.....	42
Listado de solicitudes.....	42
Revisión de solicitud.....	43
Alta de solicitud.....	44
Comprobar estado solicitud.....	44
Diagrama de flujo entre pantallas.....	45
Módulo de horario.....	45
Casos de uso.....	46
Mostrar horario.....	46
Informe horario.....	46
Alta de hora.....	47
Modificación de hora.....	47
Asignar casilla.....	47
Prototipos de interfaz gráfica.....	48
Horario.....	48
Alta de hora.....	48
Editar hora.....	49
Asignar casilla.....	49

Diagrama de flujo entre pantallas.....	50
Diseño técnico.....	50
Diseño de las clases.....	50
Persistencia.....	50
Dominio.....	51
Servicios.....	52
Negocio.....	52
Manejadores.....	53
Gestores de informe.....	53
Utilidades.....	54
Detalle de clases.....	54
ComponentesService.....	54
UsuariosService.....	55
ReunionesService.....	55
PerfilesService.....	55
GruposService.....	56
HorarioService.....	56
SolicitudesService.....	56
ReunionesAction.....	56
ComponentesAction.....	57
UsuariosAction.....	57
SolicitudesAction.....	58
PerfilesAction.....	58
GruposAction.....	58
HorarioAction.....	59
AutenticacionAction.....	59
Reports.....	59
UsuariosReport.....	60
GruposReport.....	60
ComponentesReport.....	60
HorarioReport.....	60
ReunionesReport.....	60
Utils.....	60
Mail.....	61
Report.....	61
Diseño de la persistencia.....	61
Diseño de las tablas.....	61
Diagrama E/R.....	63
Detalle de tablas.....	63
Grupos.....	63
Horas.....	64
Perfiles.....	64
Reuniones.....	65
Solicitudes.....	65
Usuarios.....	66
Asistentes.....	66
Casillas.....	67
Componentes.....	67
Arquitectura.....	67
Capas.....	68
Tecnología.....	68

Frameworks.....	68
Diagrama de componentes.....	69
Planificación.....	69
Hitos y distribución.....	69
Diagrama de Gantt.....	71
Implantación.....	71
Configuración de los recursos.....	72
Generación del archivo WAR.....	72
Configuración de LibreOffice en modo servidor.....	73
Instalación del driver JDBC de MySQL en Apache Tomcat.....	74
Despliegue de la aplicación en Apache Tomcat.....	74
Errores frecuentes.....	74
Resumen.....	75
Conclusiones.....	75
Glosario de términos.....	75

# Introducción

---

## Descripción del proyecto

Este proyecto permite la gestión de los distintos elementos que componen una asociación de grupos de música. Aunque el desarrollo estará orientado a una asociación y a un tipo de organización concretos, tendrá ciertas características que podrán ser extensibles a otro tipo de asociaciones de grupos musicales.

El proyecto abarcará la administración de los elementos de una asociación de estas características: socios/usuarios, grupos de música, componentes, horarios de ensayo, reuniones, etc. También se incluirán perfiles de usuario que permitirán acceder a ciertas opciones del menú.

En el módulo de socios/usuarios se podrán dar de alta los usuarios, modificar sus datos, asignar perfiles, asignarles grupos, generar un listado de socios, etc. No todas las acciones estarán disponibles para todos los usuarios que accedan a la aplicación, ya que eso dependerá del perfil que tenga el usuario que haya iniciado sesión.

El módulo de grupos permitirá la definición de grupos, generar un listado de grupos, añadir componentes al grupo, etc.

El módulo de componentes permitirá modificar los datos de los componentes, crear nuevos, generar un listado con los componentes, etc.

En el módulo de horarios podemos definir horas, asignar las horas de ensayo disponibles a grupos, generar el esquema con el horario actual, etc.

Podremos también almacenar las reuniones, asignar a los usuarios/socios que hayan asistido, introducir los acuerdos tomados y las discusiones realizadas, etc.

La aplicación permitirá también la realización de solicitudes de acceso a la asociación, es decir, una persona externa podrá solicitar el acceso indicando ciertos datos.

Este proyecto debe dar soporte y mantenimiento a las necesidades de la asociación de grupos musicales. Por tanto, será necesario la gestión de varias entidades identificadas:

- **Grupos.** La aplicación debe permitir dar de alta, modificar y dar de baja a los grupos de la asociación y sus componentes. Además de esto, debe permitir generar un listado para llevar un control de los grupos que están dados de alta en la asociación.
- **Usuarios.** La aplicación debe dar soporte a la gestión de usuarios (alta, baja y modificación). También permitirá generar un listado con los socios/usuarios que se encuentran en el sistema para poder llevar el acta de socios.
- **Solicitudes.** Es necesaria la gestión de solicitudes para una correcta administración de las mismas. La aplicación deberá disponer de medios que permitan registrarse como solicitante y valorar las solicitudes.
- **Horario.** El proyecto también administrará el horario del local de ensayos. Se repartirán las horas disponibles y se podrá generar un esquema con la ocupación actual de las casillas.
- **Reuniones.** La asociación celebra reuniones de las cuales es necesario generar un acta y guardar ciertos datos que la aplicación gestionará.

## Escenario

La asociación está gestionada por las mismas personas que la forman. A modo de resumen, la finalidad de la asociación es permitir a ciertas personas con bajo nivel adquisitivo la disposición de ciertos materiales y un local de ensayos donde poder realizar su actividad musical.

El volumen de personas y grupos que tiene la asociación está limitado a las casillas libres del horario, si bien es cierto que existen grupos registrados que, por problemas de tiempo, no disponen de una casilla para ensayos. Puesto que el volumen de personas es relativamente pequeño, no será necesaria la implantación de una gran estructura que permita alta disponibilidad para poder hacer frente a una gran demanda, porque ésta no se dará.

Debido a estas características, el proyecto puede disponer de ciertas operaciones que no aludan a una completa optimización (como procedimientos ejecutados con una frecuencia temporal baja, cada media hora por ejemplo), puesto que es de esperar que la base de datos no sea extensa.

## Esquema

Este proyecto constará de cuatro partes bien diferenciadas:

1. **Ordenador personal.** Cada usuario se conecta a su equipo personal y desde el mismo podrá hacer las gestiones necesarias sin necesidad de usar otro componente más que una conexión a Internet y un navegador web (Mozilla Firefox, Opera, Safari, etc).
2. **Internet.** Es un componente muy importante ya que será el que nos permita conectar con el servidor que procesará nuestras acciones y peticiones y a través del cual recibiremos la respuesta.
3. **Servidor.** Será un equipo físico o virtual el cual tendrá instalado el contenedor de Servlets que permita la ejecución de la aplicación. También tendrá instalado un servidor de conversión de documentos para la generación de la documentación.
4. **Base de datos.** El sistema gestor de base de datos nos permitirá la persistencia de los datos. Podrá estar instalado en el servidor o en otro equipo distinto que sea visible por el servidor.



## Objetivos

---

Con el trabajo fin de carrera se pretende conocer y asimilar todas las etapas por las que pasa un proyecto software desde el nacimiento de una necesidad (en nuestro caso, administrar una asociación de grupos musicales) hasta la implantación del producto software final.

En nuestro caso, tenemos la necesidad de administrar la asociación de grupos musicales "En clave de rock". Al ser una asociación de grupos musicales real, tenemos un mejor conocimiento de los elementos que vamos a administrar. Además, contamos unos clientes reales que nos indicarán la cobertura que debe dar la aplicación.

El objetivo principal será superar todas las fases del proyecto dejando un producto final que sea fiable, robusto y que satisfaga las necesidades del cliente.



# Análisis de requisitos

---

## Requerimientos

Como hemos indicado, este proyecto permitirá la gestión y administración de los distintos elementos que componen una asociación de grupos musicales. Para ello, la aplicación estará dividida en distintos módulos atendiendo a la gestión de cada uno de los elementos.

Ciertos módulos podrán tener funcionalidades solapadas en otros, si bien es cierto que, en el módulo específico, dicha funcionalidad solapada será tratada con mayor nivel de profundidad. Todos los módulos estarán controlados con perfiles, es decir, se asignarán perfiles a cada usuario de forma que sólo podrá realizar las funciones que dicho perfil permita.

## Módulo de socios/usuarios

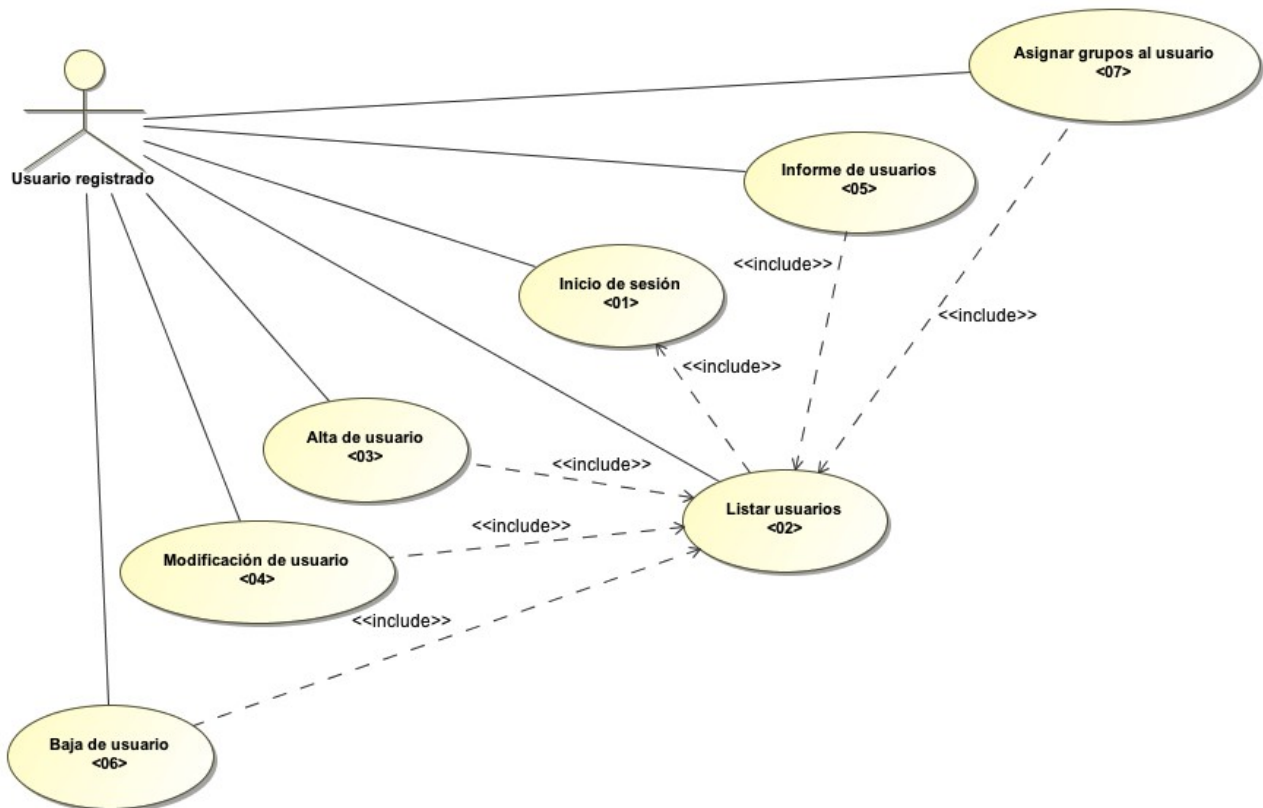
Este módulo permite la gestión de usuarios/socios de la asociación. Aquí se podrán dar de alta nuevos usuarios. Podremos asignar perfiles a los usuarios/socios. También podremos dar de baja a los usuarios que ya no estén en la asociación y modificar sus datos si fuera necesario. La aplicación también permitirá generar un listado de los socios que están en la asociación mostrando ciertos datos de interés. Aquí también podremos asignarle grupo al usuario (función solapada con el módulo de componentes).

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Existirá un tipo de usuario Administrador que contará con perfil Administrador y que permitirá el uso avanzado de ciertas características limitadas para otros usuarios
- El usuario estará referido al usuario del sistema y al socio registrado. Los administradores que formen parte de la asociación deberán tener una cuenta aparte como usuarios registrados y usar la primera sólo para tareas de mantenimiento y administración
- Cada usuario deberá tener un único perfil que podrá variar en el tiempo
- Un usuario sin perfil se considera no registrado
- Las tareas de mantenimiento de usuarios (alta, modificación, baja, etc) estarán en manos de los Administradores y los usuarios que tengan un perfil que les permita realizar dichas tareas
- Los usuarios deben ser informados del alta o la modificación de algún dato relevante (el usuario y contraseña) mediante correo electrónico si se produjera tal caso
- Los usuarios no se eliminarán del sistema. Se les asignará una fecha de baja, ya que podría volver a incorporarse
- Los usuarios deberán aportar algunos datos de forma obligatoria que permita la localización y el contacto (correo electrónico)
- Los usuarios pueden registrarse con fechas de alta futuras o pasadas (traspaso de ficheros)
- Pueden existir usuarios de nacionalidad no española

Como asociación, es obligatorio mantener un registro de socios. La aplicación deberá disponer de una opción que permita elaborar un listado en formato PDF donde poder ver el registro actual de socios.

## Casos de uso



### Inicio de sesión

Código	<01>
Resumen	El usuario inicia sesión en el sistema introduciendo su usuario y clave.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder acceder a las funcionalidades del sistema.
Actores	Usuario registrado
Relacionados	-
Flujo normal	<ol style="list-style-type: none"> <li>1. Se introduce el usuario del sistema</li> <li>2. Se introduce la clave para ese usuario</li> <li>3. Se accede al menú principal del sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario o/y la clave son incorrectos</li> <li>• El usuario no tiene permisos para acceder</li> </ul>

### Listar usuarios

Código	<02>
Resumen	Se muestra un listado con los usuarios que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar los usuarios que existen en la base de datos.

Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Registro de socios"</li> <li>3. Se muestra el listado</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>

### Alta de usuario

Código	<03>
Resumen	Damos de alta a un usuario en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder insertar los datos de un nuevo usuario en el sistema.
Actores	Usuario registrado
Relacionados	Listar usuarios <02>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;02&gt;</li> <li>2. Se pulsa el botón "Nuevo usuario"</li> <li>3. Se insertan los datos solicitados y se pulsa el botón "Guardar"</li> <li>4. El usuario se almacena en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El DNI de usuario ya existe en el sistema</li> <li>• El correo electrónico ya existe en el sistema</li> <li>• El campo usuario ya existe en el sistema</li> <li>• No existen perfiles disponibles en la base de datos</li> </ul>

### Modificación de usuario

Código	<04>
Resumen	Modificamos los datos de un usuario del sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos de un usuario registrado en el sistema.
Actores	Usuario registrado
Relacionados	Listar usuarios <02>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;02&gt;</li> <li>2. Se pulsa el botón "Editar" del usuario deseado</li> <li>3. Se editan los datos del usuario y se pulsa el botón "Guardar cambios"</li> <li>4. Se guardan los cambios en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El DNI de usuario ya existe en el sistema</li> <li>• El correo electrónico ya existe en el sistema</li> <li>• El campo usuario ya existe en el sistema</li> </ul>

### Informe de usuarios

Código	<05>
Resumen	Obtenemos un informe en PDF con los usuarios registrados en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder obtener el informe de los usuarios que están registrados en el sistema.
Actores	Usuario registrado
Relacionados	Listar usuarios <02>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;02&gt;</li> <li>2. Se pulsa el botón "Exportar en PDF"</li> <li>3. Obtenemos el listado de usuarios</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Listado vacío</li> <li>• Servidor de conversión PDF desconectado</li> </ul>

### Baja de usuario

Código	<06>
Resumen	Asignamos una fecha de baja al usuario para que deje de estar vigente.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder dar de baja a un usuario en el sistema.
Actores	Usuario registrado
Relacionados	Listar usuarios <02>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;02&gt;</li> <li>2. Se pulsa el botón "Editar" del usuario deseado</li> <li>3. Se añade la fecha de baja y se pulsa el botón "Guardar cambios"</li> <li>4. El usuario deja de estar vigente en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El usuario ya está dado de baja</li> <li>• La fecha de baja es anterior a la fecha de alta</li> </ul>

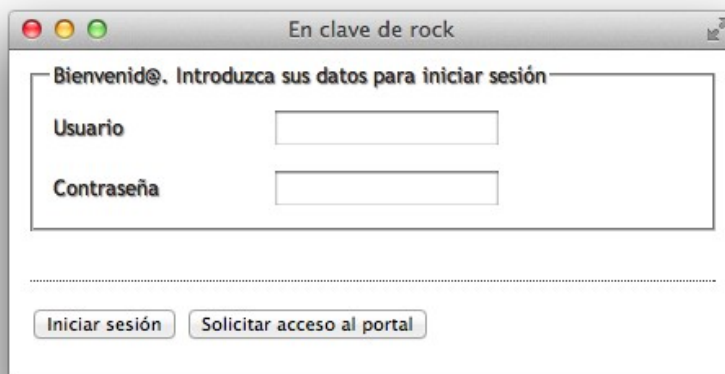
### Asignar grupo al usuario

Código	<07>
Resumen	Asignamos un grupo al usuario que estamos visualizando.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder asignar un grupo al usuario que esté editando.
Actores	Usuario registrado
Relacionados	Listar usuarios <02>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;02&gt;</li> <li>2. Se pulsa el botón "Editar" del usuario deseado</li> <li>3. Se pulsa el botón "Asignar grupo..."</li> <li>4. Se selecciona el grupo y se introducen los datos</li> <li>5. El usuario queda asignado al grupo como componente</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• No existen grupos para el usuario deseado</li> </ul>

## Prototipos de interfaz gráfica

### Inicio de sesión

En esta pantalla, el usuario registrado insertará su usuario y clave para iniciar sesión en la plataforma. Desde aquí también se podrá acceder a la solicitud de acceso al portal para los usuarios no registrados (módulo de solicitudes).



Enclave de rock

Bienvenid@. Introduzca sus datos para iniciar sesión

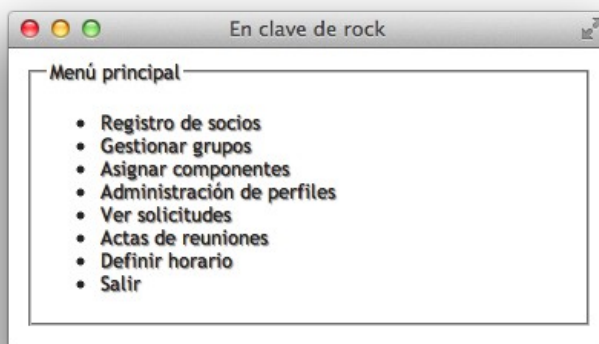
Usuario

Contraseña

Iniciar sesión Solicitar acceso al portal

### Menú principal

El usuario registrado verá esta pantalla después de haber iniciado sesión correctamente. Podrá seleccionar algunas o todas las opciones del menú dependiendo de los permisos que tenga. Las opciones visibles, serán aquellas en las que el perfil de dicho usuario tenga permisos de creación, edición o baja.



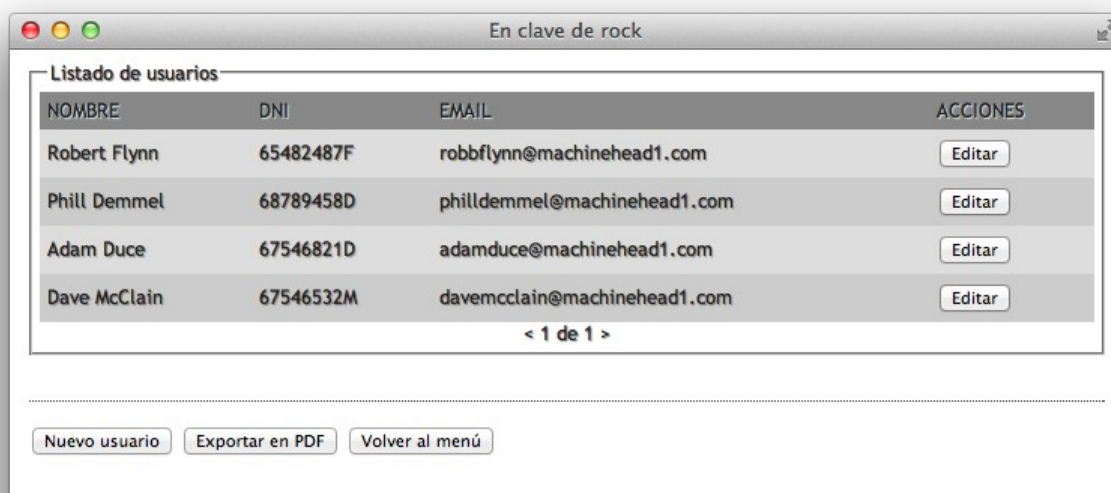
Enclave de rock

Menú principal

- Registro de socios
- Gestionar grupos
- Asignar componentes
- Administración de perfiles
- Ver solicitudes
- Actas de reuniones
- Definir horario
- Salir

### Listado de usuarios

Se mostrarán los usuarios vigentes o no vigentes según los permisos del usuario registrado que haya iniciado sesión. Desde esta pantalla también se permite obtener el informe de usuarios a partir del botón "Exportar en PDF" y obtendremos el listado de usuarios activos a fecha de hoy (exceptuando los administradores).



### Alta de usuario

Esta pantalla permitirá el alta de usuario en el sistema. Tendrá algunos campos obligatorios (correo electrónico, perfil, usuario, etc) y otros opcionales por si se quisiera detallar la información del nuevo usuario. La contraseña no será un dato obligatorio de forma que, si el usuario registrado que esté realizando el alta no introduce una contraseña, el sistema la generará y se le enviará al usuario a registrar mediante correo electrónico.

Una vez dado de alta, el sistema podrá enviar un correo electrónico al nuevo usuario (según la dirección que se haya introducido) indicándole sus datos de acceso y pidiéndole que revise sus datos por si hubiera algún dato erróneo.

Desde esta pantalla podemos volver al listado de usuarios o al menú principal usando los botones de navegación situados en la parte inferior.

The screenshot shows a web application window titled "En clave de rock". The window contains a form for editing user data, organized into three main sections:

- Datos del usuario:** Includes fields for DNI, Nombre, Primer apellido, Segundo apellido, Fecha de nacimiento, and Sexo (a dropdown menu).
- Datos de contacto:** Includes fields for Domicilio, Código postal, Localidad, Provincia, País, Correo electrónico, Teléfono móvil, and Teléfono fijo.
- Otros datos:** Includes fields for Usuario, Contraseña, Fecha de alta, Fecha de baja, Perfil (a dropdown menu), Web, Perfil en Facebook, and Perfil en Twitter.

At the bottom of the form, there are three buttons: "Guardar", "Volver al listado", and "Volver al menú".

### Modificación de usuario

En esta pantalla podremos editar los datos de un usuario que ya se encuentre en el sistema. A esta pantalla accedemos mediante la opción "Editar" de la pantalla con el listado de usuarios.

Se podrán editar todos los datos del usuario siempre que el perfil del usuario registrado que esté editando lo permita. También se podrá cambiar el perfil asignado o añadir una fecha de baja para que el usuario que estamos editando deje de estar vigente. Además de esto, podremos asignarle grupos para crear componentes.

Desde aquí también podemos volver al listado de usuarios o bien al menú principal.

The screenshot shows a web application window titled "En clave de rock". The window contains a form for editing a user's profile, organized into three main sections:

- Datos del usuario:**
  - DNI: 65482487F
  - Nombre: Robert
  - Primer apellido: Flynn
  - Segundo apellido: (empty)
  - Fecha de nacimiento: 19/07/1968
  - Sexo: MASCULINO (dropdown menu)
- Datos de contacto:**
  - Domicilio: Madison Street, 20th
  - Código postal: 2009
  - Localidad: Oakland
  - Provincia: California
  - País: USA
  - Correo electrónico: robbflynn@machinehe:
  - Teléfono móvil: (empty)
  - Teléfono fijo: (empty)
- Otros datos:**
  - Usuario: robbflynn
  - Contraseña: (masked with dots)
  - Fecha de alta: 10/08/2005
  - Fecha de baja: (empty)
  - Perfil: Presidente (dropdown menu)
  - Web: www.robbflynn.com
  - Perfil en Facebook: (empty)
  - Perfil en Twitter: @thegeneralmh

At the bottom of the form, there are four buttons: "Guardar cambios", "Asignar grupo...", "Volver al listado", and "Volver al menú".

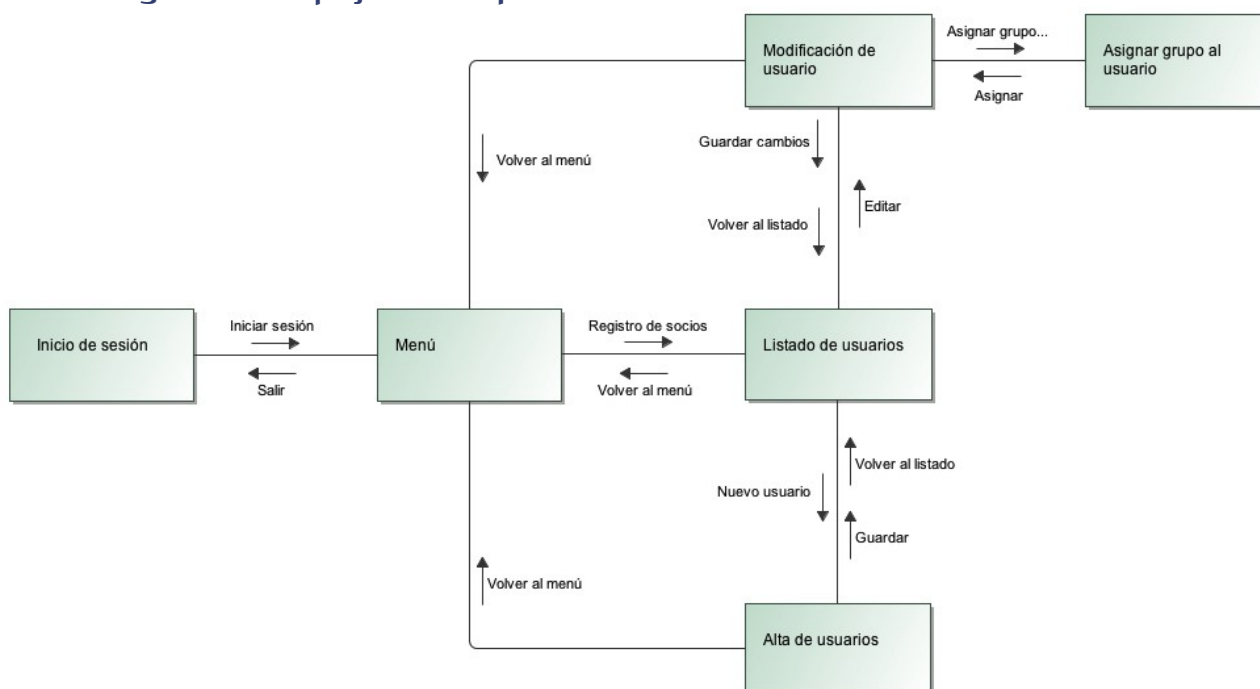
### Asignar grupo al usuario

Accedemos a esta pantalla usando la opción "Asignar grupo..." de la pantalla de edición de usuarios. Aquí se introducen los datos del usuario en el grupo como componente. También podemos asignar una fecha de salida del grupo dependiendo del perfil que tenga el usuario registrado que esté asignando el grupo.

La actividad desempeñada en el grupo será un campo abierto, puesto que las posibles opciones son demasiado amplias y abiertas como para limitarlas a una lista cerrada. El campo representante indica que el usuario que estamos editando es el representante del grupo destino.



## Diagrama de flujo entre pantallas



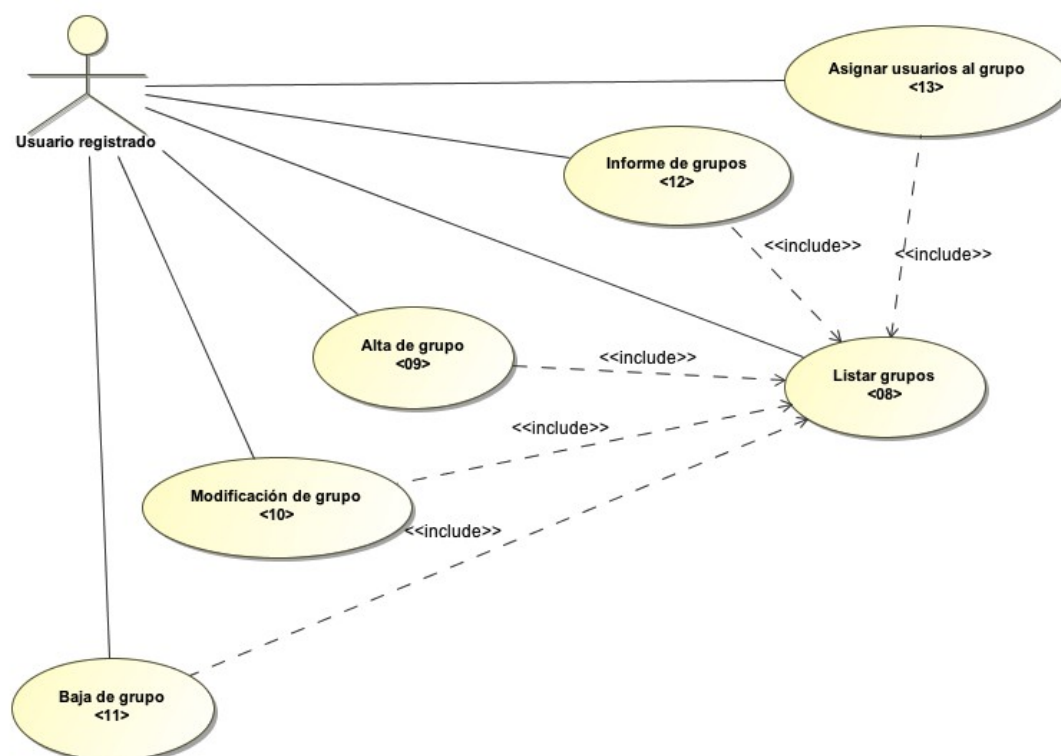
## Módulo de grupos

Este módulo permite la gestión de los grupos de la asociación. Podremos crear nuevos grupos indicando ciertos datos relevantes de cada uno de ellos. La aplicación también permitirá la generación del listado de grupos de la asociación. También existirá la posibilidad de dar de baja al grupo de la asociación o asignarle componentes (función solapada con el módulo de componentes).

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Cada grupo debe poder diferenciarse sólo por el nombre del mismo
- Los grupos puede tener un representante como máximo
- Los grupos pueden tener  $n$  componentes
- Los grupos podrán tener diferentes tendencias y estilos
- Los grupos podrán tener fecha de inscripción futura o pasada (traspaso de ficheros)

## Casos de uso



Listar grupos	
Código	<08>
Resumen	Se muestra un listado con los grupos que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar los grupos que existen en la base de datos.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Gestionar grupos"</li> <li>3. Se muestra el listado</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> </ul>

- El listado está vacío

### Alta de grupo

Código	<09>
Resumen	Damos de alta a un grupo en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder insertar los datos de un nuevo grupo en el sistema.
Actores	Usuario registrado
Relacionados	Listar grupos <08>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;08&gt;</li> <li>2. Se pulsa el botón "Nuevo grupo"</li> <li>3. Se insertan los datos solicitados y se pulsa el botón "Guardar"</li> <li>4. El grupo se almacena en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> <li>• Los datos tienen un formato incorrecto</li> <li>• El grupo ya existe en el sistema</li> </ul>

### Modificación de grupo

Código	<10>
Resumen	Modificamos los datos de un grupo del sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos de un grupo registrado en el sistema.
Actores	Usuario registrado
Relacionados	Listar grupos <08>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;08&gt;</li> <li>2. Se pulsa el botón "Editar" del grupo deseado</li> <li>3. Se editan los datos del grupo y se pulsa el botón "Guardar cambios"</li> <li>4. Se guardan los cambios en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> <li>• Los datos tienen un formato incorrecto</li> </ul>

### Baja de grupo

Código	<11>
Resumen	Asignamos una fecha de baja al grupo para que deje de estar vigente.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder dar de baja a un grupo en el sistema.
Actores	Usuario registrado
Relacionados	Listar grupos <08>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;08&gt;</li> <li>2. Se pulsa el botón "Editar" del grupo deseado</li> <li>3. Se añade la fecha de baja y se pulsa el botón "Guardar cambios"</li> </ol>

Flujo alternativo/excepciones
-------------------------------

4. El grupo deja de estar vigente en el sistema
  - El usuario no tiene permisos para realizar esta acción
  - El grupo ya está dado de baja
  - La fecha de baja es anterior a la fecha de alta

Informe de grupo	
Código	<12>
Resumen	Obtenemos un informe en PDF con los grupos registrados en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder obtener el informe de los grupos que están registrados en el sistema.
Actores	Usuario registrado
Relacionados	Listar grupos <08>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;08&gt;</li> <li>2. Se pulsa el botón "Exportar en PDF"</li> <li>3. Obtenemos el listado de grupos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Listado vacío</li> <li>• Servidor de conversión PDF desconectado</li> </ul>

Asignar usuarios al grupo	
Código	<13>
Resumen	Asignamos un usuarios al grupo que estamos visualizando.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder asignar componentes al grupo que esté editando.
Actores	Usuario registrado
Relacionados	Listar grupos <08>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;08&gt;</li> <li>2. Se pulsa el botón "Editar" del grupo deseado</li> <li>3. Se pulsa el botón "Asignar componentes..."</li> <li>4. Se seleccionan los usuarios y se introducen los datos</li> <li>5. Los usuarios quedan asignados al grupo como componentes</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• No existen usuarios</li> </ul>

## Prototipos de interfaz gráfica

### Listado de grupos

Se mostrarán los grupos vigentes o no vigentes según los permisos del usuario registrado que haya iniciado sesión. Desde esta pantalla también se permite obtener el informe de grupos a partir del botón "Exportar en PDF" y obtendremos un listado con algunos datos de cada grupo y representante.



### Alta de grupo

Esta pantalla permitirá el alta de grupo en el sistema. Contiene los datos básicos de un grupo y permite introducir otros datos que se quisieran detallar, como la web, el correo electrónico, etc.

Desde esta pantalla podemos volver al listado de grupos o al menú principal usando los botones de navegación situados en la parte inferior.



### Modificación de grupo

En esta pantalla podremos editar los datos de un grupo que ya se encuentre en el sistema. A esta pantalla accedemos mediante la opción "Editar" de la pantalla con el listado de grupos.

Aquí se editan todos los datos del grupo siempre que el perfil del usuario registrado que esté editando lo permita. Podremos añadir una fecha de baja para que el grupo deje de estar vigente o bien asignar usuarios para crear componentes del grupo.

Desde aquí también podemos volver al listado de grupos o bien al menú principal.

### Asignar usuarios al grupo

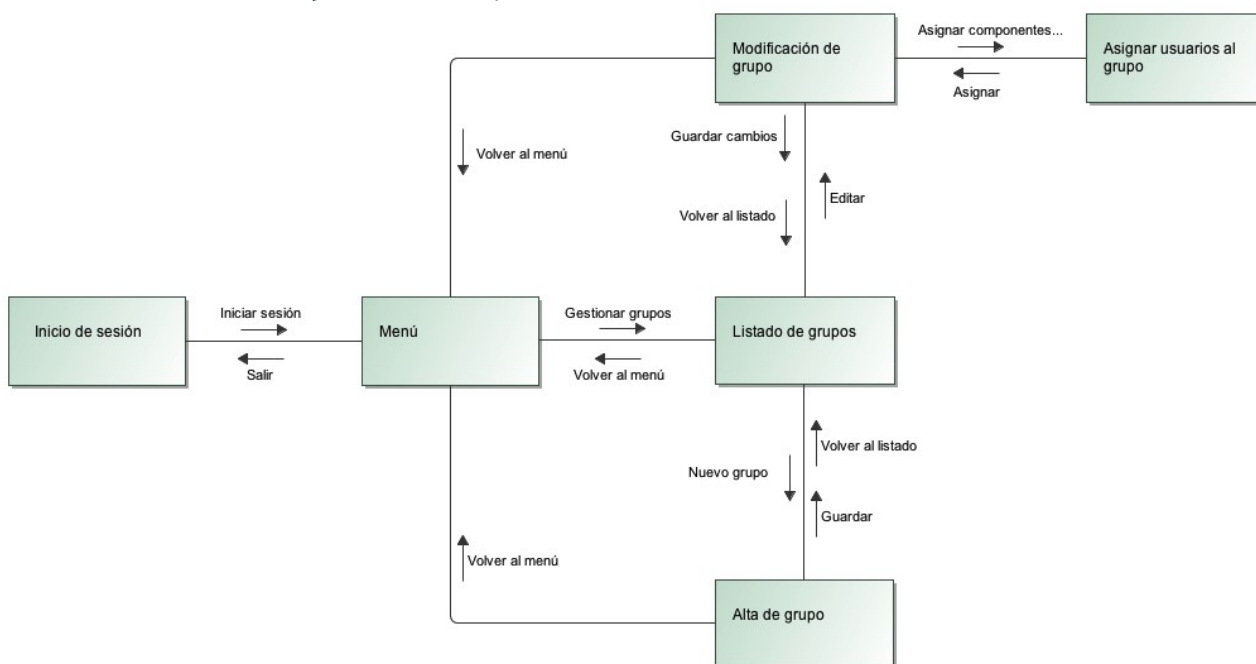
Accedemos a esta pantalla usando la opción "Asignar componentes..." de la pantalla de modificación de grupo. Aquí se seleccionan los usuarios que van a formar parte del grupo, indicando la actividad desempeñada en el grupo y otros datos. También podemos asignar una fecha de salida del componente dependiendo del perfil que tenga el usuario registrado que esté editando.

NOMBRE	FECHA DE ENTRADA	FECHA DE SALIDA	ACTIVIDAD	REPRESENTANTE	SELECCIONAR
Robert Flynn	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Phill Demmel	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Adam Duce	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dave McClain	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

< 1 de 1 >

Asignar

## Diagrama de flujo entre pantallas



## Módulo de perfiles

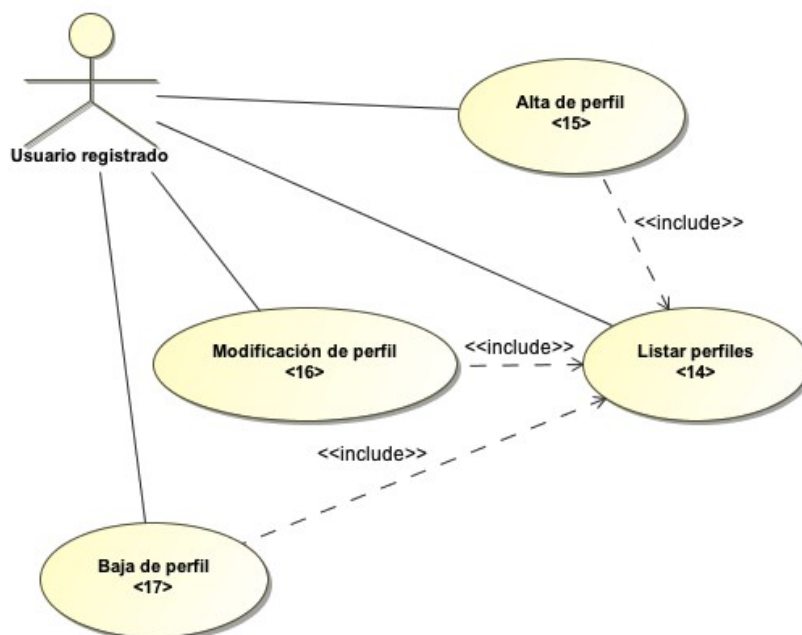
Este módulo permite la gestión de los perfiles de usuario. En este apartado, podremos crear los perfiles e indicar dónde pueden y no pueden acceder los usuarios. Podremos modificar o/y dar de baja los perfiles que ya no usemos.

Aunque sea uno de los módulos más simples en lo que a volumen de casos de uso se refiere, hay que destacar que es uno de los apartados más importantes del sistema, puesto que modifica la tabla diccionario que dictará lo que un usuario podrá y no podrá hacer en la aplicación.

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Los perfiles deben tener nombres diferentes, aún cuando éstos estén dados de baja
- Los perfiles deberán determinar lo que un usuario puede y no puede manipular en la aplicación
- Todos los usuarios deben tener un perfil y sólo uno al mismo tiempo
- Los perfiles pueden asignarse a  $n$  usuarios
- Los perfiles pueden modificarse y tener distintos permisos en el tiempo incluso pueden causar baja

## Casos de uso



### Listar perfiles

Código	<14>
Resumen	Se muestra un listado con los perfiles que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar los perfiles que existen en la base de datos.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Administración de perfiles"</li> <li>3. Se muestra el listado de perfiles</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>

### Alta de perfil

Código	<15>
Resumen	Damos de alta un perfil en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder insertar los datos de un nuevo perfil en el sistema.
Actores	Usuario registrado
Relacionados	Listar perfiles <14>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;14&gt;</li> <li>2. Se pulsa el botón "Nuevo perfil"</li> <li>3. Se insertan los datos solicitados y se pulsa el botón "Guardar"</li> </ol>



Flujo alternativo/excepciones
-------------------------------

4. El perfil se almacena en la base de datos
  - El usuario no tiene permisos para realizar esta acción
  - Faltan datos por rellenar
  - El perfil ya existe en el sistema

### Modificación de perfil

Código	<16>
Resumen	Modificamos los datos de un perfil del sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos de un perfil registrado en el sistema.
Actores	Usuario registrado
Relacionados	Listar perfiles <14>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;14&gt;</li> <li>2. Se pulsa el botón "Editar" del perfil deseado</li> <li>3. Se editan los datos del perfil y se pulsa el botón "Guardar cambios"</li> <li>4. Se guardan los cambios en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> </ul>

### Baja de perfil

Código	<17>
Resumen	Asignamos una fecha de baja al perfil para que deje de estar vigente.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder dar de baja a un perfil en el sistema.
Actores	Usuario registrado
Relacionados	Listar perfiles <14>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;14&gt;</li> <li>2. Se pulsa el botón "Editar" del perfil deseado</li> <li>3. Se añade la fecha de baja y se pulsa el botón "Guardar cambios"</li> <li>4. El perfil deja de estar vigente en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El perfil ya está dado de baja</li> <li>• La fecha de baja es anterior a la fecha de alta</li> </ul>

## Prototipos de interfaz gráfica

### Listado de perfiles

Se mostrarán los perfiles vigentes o no vigentes según los permisos del usuario registrado que haya iniciado sesión.



### Alta de perfil

Esta pantalla permitirá el alta de perfil en el sistema. Contiene el nombre del perfil y los permisos que lleva asociado este perfil para indicar las acciones que puede llevar a cabo un usuario con el mismo.

Desde esta pantalla podemos volver al listado de perfiles o al menú principal usando los botones de navegación situados en la parte inferior.

En clave de rock

Datos del perfil

Nombre  Fecha de alta

Fecha de baja

Los usuarios con este perfil pueden

- Crear usuarios
- Eliminar usuarios
- Modificar perfiles
- Asignar perfiles
- Modificar componentes
- Crear grupos
- Eliminar grupos
- Modificar reuniones
- Crear horas
- Eliminar horas
- Modificar solicitudes
- Modificar usuarios
- Crear perfiles
- Eliminar perfiles
- Crear componentes
- Eliminar componentes
- Modificar grupos
- Crear reuniones
- Eliminar reuniones
- Modificar horas
- Ver solicitudes
- Eliminar solicitudes

Guardar Volver al listado Volver al menú

### Modificación de perfil

En esta pantalla podremos editar los datos de un perfil que ya se encuentre en el sistema. A esta pantalla accedemos mediante la opción "Editar" de la pantalla con el listado de perfiles.

Aquí se editan los datos del perfil siempre que el perfil del usuario registrado que esté editando lo permita. También podremos añadir una fecha de baja para que el perfil deje de estar vigente.

Desde aquí podemos volver al listado de perfiles o bien al menú principal.

En clave de rock

---

**Datos del perfil**

Nombre	<input type="text" value="Presidente"/>	Fecha de alta	<input type="text" value="10/03/2008"/>
Fecha de baja	<input type="text"/>		

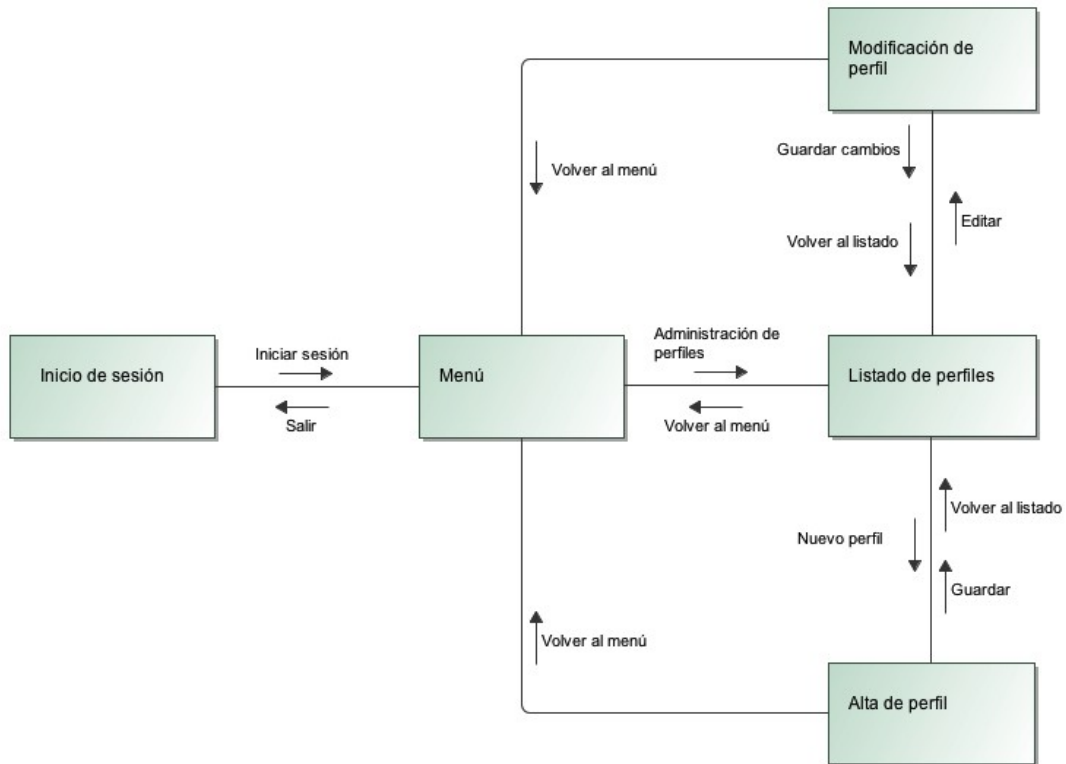
---

**Los usuarios con este perfil pueden**

<input checked="" type="checkbox"/> Crear usuarios	<input checked="" type="checkbox"/> Modificar usuarios
<input checked="" type="checkbox"/> Eliminar usuarios	<input type="checkbox"/> Crear perfiles
<input type="checkbox"/> Modificar perfiles	<input type="checkbox"/> Eliminar perfiles
<input checked="" type="checkbox"/> Asignar perfiles	<input checked="" type="checkbox"/> Crear componentes
<input checked="" type="checkbox"/> Modificar componentes	<input checked="" type="checkbox"/> Eliminar componentes
<input checked="" type="checkbox"/> Crear grupos	<input checked="" type="checkbox"/> Modificar grupos
<input checked="" type="checkbox"/> Eliminar grupos	<input checked="" type="checkbox"/> Crear reuniones
<input checked="" type="checkbox"/> Modificar reuniones	<input checked="" type="checkbox"/> Eliminar reuniones
<input checked="" type="checkbox"/> Crear horas	<input checked="" type="checkbox"/> Modificar horas
<input checked="" type="checkbox"/> Eliminar horas	<input checked="" type="checkbox"/> Ver solicitudes
<input checked="" type="checkbox"/> Modificar solicitudes	<input checked="" type="checkbox"/> Eliminar solicitudes

---

## Diagrama de flujo entre pantallas



## Módulo de componentes

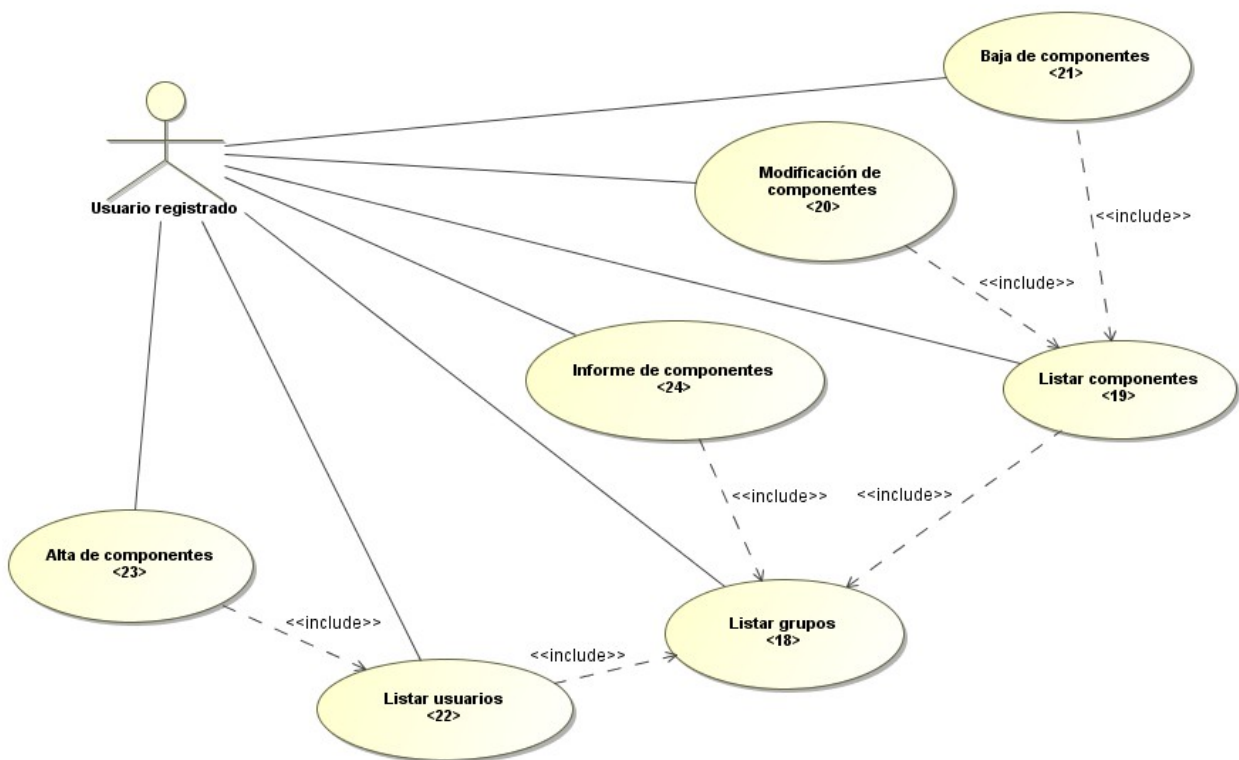
Este módulo permite la gestión de los componentes de la asociación. Aquí se podrán crear, modificar y dar de baja componentes. Además, se permitirá generar un informe con los componentes de cada grupo.

Este módulo tiene funcionalidades que ya se han implementado en otros módulos, si bien es cierto que desde éste se hace una gestión algo más avanzada que en los demás.

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Los grupos pueden tener  $n$  componentes
- Un usuario puede ser componente en  $n$  grupos realizando la misma o distinta actividad en cada uno de ellos

## Casos de uso



### Listar grupos

Código	<18>
Resumen	Se muestra un listado con los grupos que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso previo que el usuario ha de seguir para poder comprobar los componentes que existen en la base de datos.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Asignar componentes"</li> <li>3. Se muestra el listado</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>
Observaciones	Nótese que este caso de uso no es igual al <08> ya que tiene un uso distinto y el contenido a mostrar será ligeramente diferente.

### Listar componentes

Código	<19>
Resumen	Listamos los componentes del grupo deseado.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder ver los datos de los componentes del grupo.
Actores	Usuario registrado

Relacionados	Listar grupos <18>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;18&gt;</li> <li>2. Se pulsa el botón "Ver/Editar componentes"</li> <li>3. Se muestran los componentes y sus datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• No existen grupos</li> <li>• No existen componentes para el grupo seleccionado</li> </ul>

### Modificación de componentes

Código	<20>
Resumen	Modificamos los datos del/de los componente/s del grupo.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos del/de los componente/s del grupo.
Actores	Usuario registrado
Relacionados	Listar componentes <19>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;19&gt;</li> <li>2. Se modifican los datos del/de los componente/s y se pulsa el botón "Guardar cambios"</li> <li>3. Se guardan los cambios en la base de datos</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> <li>• El grupo no tiene componentes</li> </ul>

### Baja de componentes

Código	<21>
Resumen	Asignamos una fecha de salida del grupo al/a los componente/s para que deje/n de estar vigente/s.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder dar de baja al/a los componente/s del grupo seleccionado.
Actores	Usuario registrado
Relacionados	Listar componentes <19>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;19&gt;</li> <li>2. Se añade la fecha de salida del grupo en el/los componente/s deseado/s y se pulsa el botón "Guardar cambios"</li> <li>3. El /Los componente/s dejan de estar vigente/s en el grupo</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El/Los componente/s ya está/n dado/s de baja</li> <li>• La fecha de baja es anterior a la fecha de alta</li> <li>• El grupo no tiene componentes</li> </ul>

### Listar usuarios

Código	<22>
Resumen	Se muestra un listado con los usuarios que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar los usuarios que existen en la base de

	datos.
Actores	Usuario registrado
Relacionados	Listar grupos <18>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;18&gt;</li> <li>2. Pulsamos "Asignar nuevos componentes" en el grupo deseado</li> <li>3. Se muestra el listado</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>
Observaciones	Nótese que no es el mismo caso de uso que el <02> ya que aquí se listarán sólo los usuarios que puedan ser componentes del grupo seleccionado

### Alta de componentes

Código	<23>
Resumen	Asignamos un componente/componentes al grupo
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder asignar componentes a un grupo seleccionado previamente.
Actores	Usuario registrado
Relacionados	Listar usuarios <22>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;22&gt;</li> <li>2. Se seleccionan el/los componente/s y se rellenan los datos</li> <li>3. Se pulsa el botón "Guardar"</li> <li>4. El/Los componente/s quedan asignados al grupo</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado de usuarios está vacío</li> <li>• Faltan datos por completar</li> </ul>

### Informe de componentes

Código	<24>
Resumen	Obtenemos un informe en PDF con los componentes registrados en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder obtener el informe de los componentes que están registrados en el sistema.
Actores	Usuario registrado
Relacionados	Listar grupos <18>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;18&gt;</li> <li>2. Se pulsa el botón "Exportar en PDF"</li> <li>3. Obtenemos el listado de componentes</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Listado vacío</li> <li>• Servidor de conversión PDF desconectado</li> </ul>



## Prototipos de interfaz gráfica

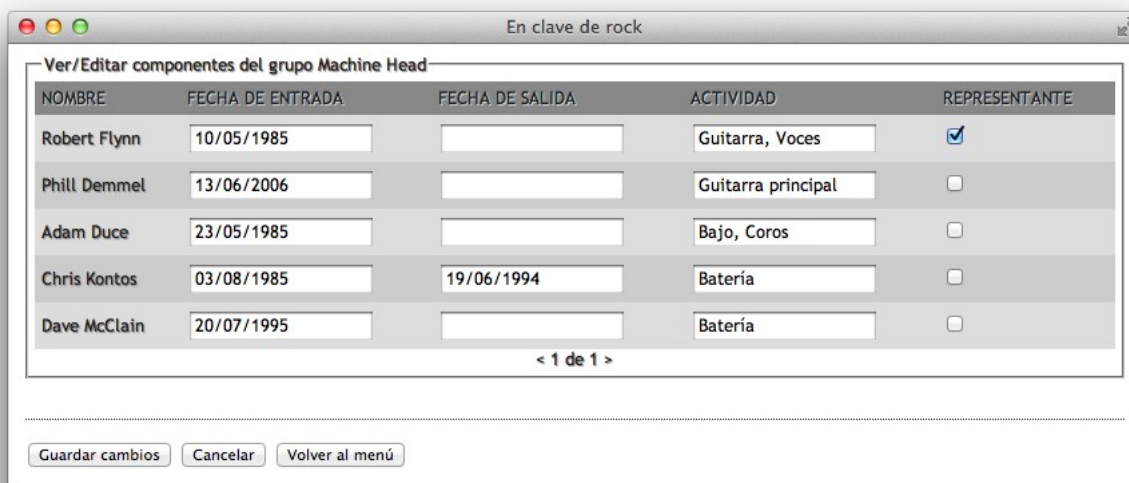
### Listado de grupos

Esta pantalla muestra el listado de los grupos que están almacenados en la base de datos. Mostrará los grupos que son susceptibles de tener componentes. Es la pantalla previa al mantenimiento completo de los componentes.



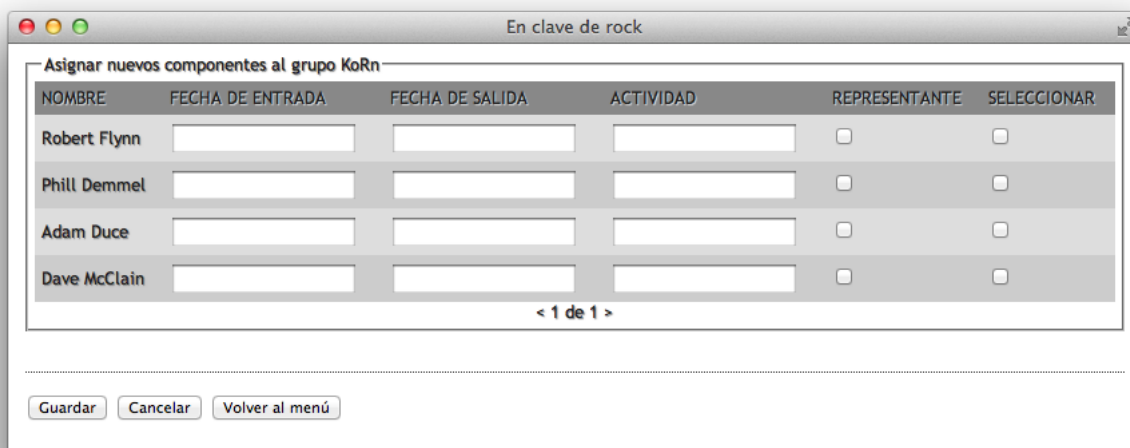
### Listado de componentes

Esta pantalla aparece al seleccionar un grupo y muestra los componentes del mismo. En ella se pueden editar los datos del componente o bien indicar la fecha de salida del grupo.

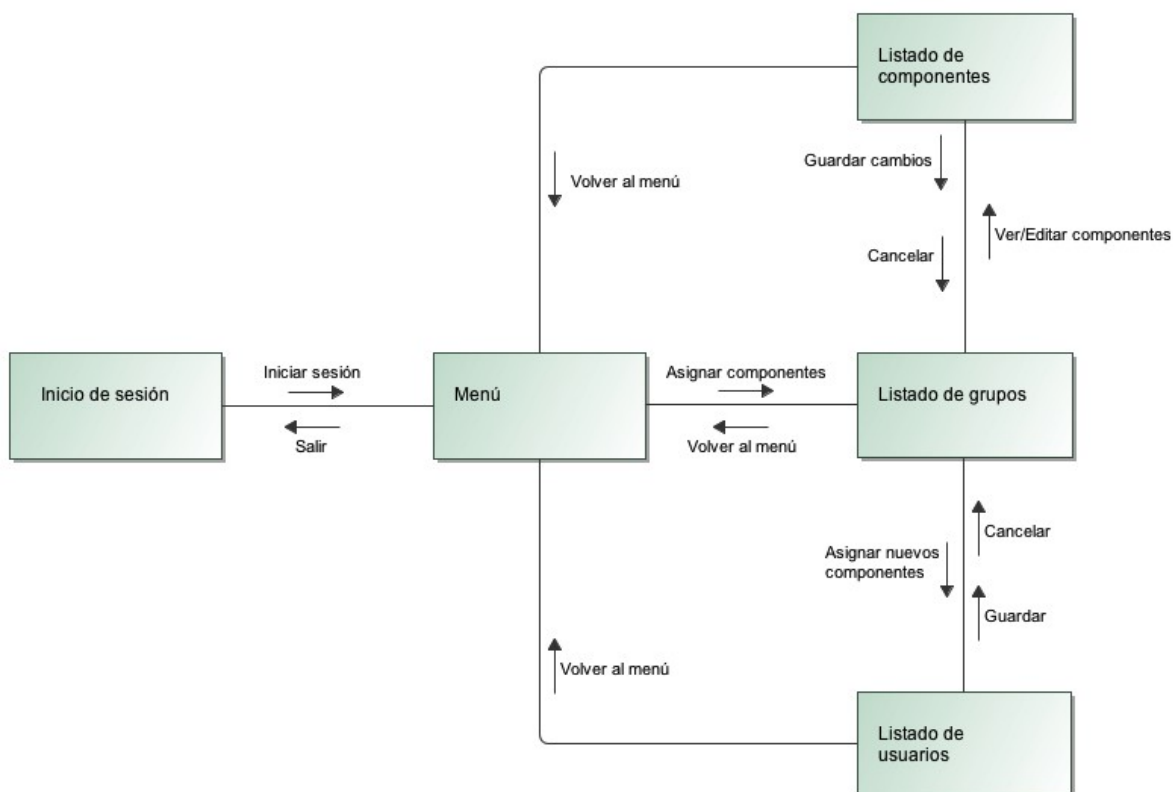


### Listado de usuarios

Aquí se muestra el listado de los usuarios que pueden ser componentes del grupo que previamente hemos seleccionado. En ella debemos seleccionar el/los usuario/s para asignarlos al grupo, indicando la fecha de entrada y la actividad desempeñada en el grupo.



## Diagrama de flujo entre pantallas



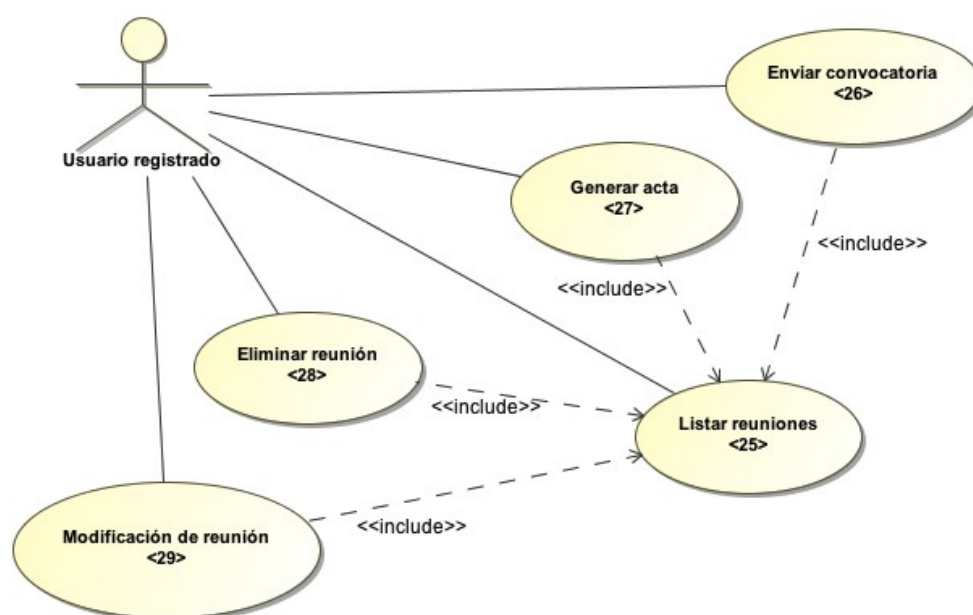
## Módulo de reuniones

Este módulo permite la gestión de las reuniones de la asociación. En este apartado, podremos almacenar los contenidos tomados en las reuniones y generar el posterior acta de reunión. También podremos asignar los componentes que asisten a las reuniones y rellenar los asuntos tratados y las decisiones y acuerdos tomados.

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Las reuniones están compuestas por una fecha, una hora, un lugar y un orden del día. Se podrán añadir acuerdos, comentarios y asistentes
- Las reuniones deben tener fecha futura
- Los usuarios deben ser notificados con la convocatoria de reunión
- Los usuarios deben ser notificados con la cancelación de la reunión si previamente ha sido convocada
- Las reuniones tienen acta
- Las reuniones notificadas (convocatoria enviada) no podrán alterar su orden del día, fecha, hora ni lugar

## Casos de uso



Listar reuniones	
Código	<25>
Resumen	Se muestra un listado con las reuniones que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar el registro de las reuniones que existen en la base de datos.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Actas de reuniones"</li> <li>3. Se muestra el listado de reuniones</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>

### Enviar convocatoria

Código	<26>
Resumen	Se envía un email a todos los usuarios interesados.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder enviar un correo electrónico con los datos de la reunión a los usuarios que puedan estar interesados en asistir a la reunión.
Actores	Usuario registrado
Relacionados	Listar reuniones <25>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;25&gt;</li> <li>2. Se pulsa el botón "Enviar convocatoria" de la reunión deseada</li> <li>3. Se envía el correo electrónico con la convocatoria a los usuarios</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> <li>• No existen posibles usuarios</li> <li>• Existen problemas en el servidor de correo electrónico</li> </ul>

### Generar acta

Código	<27>
Resumen	Se genera un informe con los datos de la reunión.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder generar el documento que sirve de acta de reunión.
Actores	Usuario registrado
Relacionados	Listar reuniones <25>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;25&gt;</li> <li>2. Se pulsa el botón "Generar acta" de la reunión deseada</li> <li>3. Obtenemos el acta de la reunión</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> <li>• Existen problemas en el servidor de conversión a PDF</li> </ul>

### Eliminar reunión

Código	<28>
Resumen	Se elimina la reunión y se notifica a los usuarios.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder eliminar una reunión y notificar a los usuarios de la cancelación de la reunión.
Actores	Usuario registrado
Relacionados	Listar reuniones <25>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;25&gt;</li> <li>2. Se pulsa el botón "Eliminar" de la reunión deseada</li> <li>3. La reunión deja de ser accesible</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>

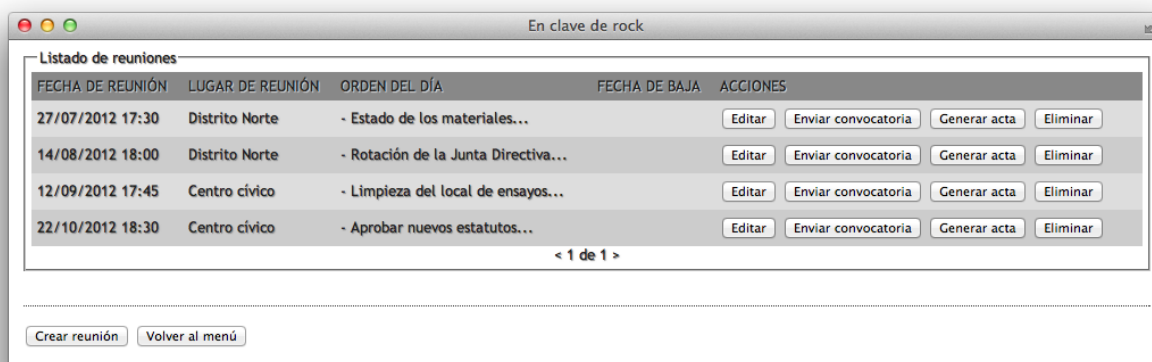
- Existen problemas en el servidor de correo electrónico

Modificación de reunión	
Código	<29>
Resumen	Se modifican los datos de la reunión deseada.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos de una reunión previamente almacenada.
Actores	Usuario registrado
Relacionados	Listar reuniones <25>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;25&gt;</li> <li>2. Se pulsa el botón "Editar" de la reunión deseada</li> <li>3. Modificamos los datos de la reunión y pulsamos el botón "Guardar cambios"</li> <li>4. Los cambios se guardan en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> <li>• Faltan datos por rellenar</li> </ul>

## Prototipos de interfaz gráfica

### Listado de reuniones

En esta pantalla se muestra el listado de las reuniones almacenadas en el sistema. Desde aquí se podrán editar, enviar la convocatoria, eliminar o generar el acta e reunión.



### Alta/Modificación de reunión

Desde esta pantalla podremos dar de alta una reunión. Para ello debemos indicar la fecha, la hora y el lugar de reunión además de un orden del día. Esta pantalla será la que veamos en la parte de modificación hasta que enviemos la convocatoria. A partir de ahí, no se podrá editar, se añadirán otros datos.

The screenshot shows a window titled "En clave de rock" with a standard Mac OS-style title bar (red, yellow, green buttons). The main content area is divided into two sections:

- Datos de la reunión:** This section contains three input fields:
  - "Fecha de la reunión": A single-line text input field.
  - "Lugar de la reunión": A single-line text input field.
  - "Hora de la reunión": A time selection control consisting of two dropdown menus separated by a colon.
- Orden del día:** A large, empty rectangular text area for entering the meeting agenda.

At the bottom of the window, there is a horizontal line followed by three buttons: "Guardar", "Volver al listado", and "Volver al menú".

#### Añadir datos/Modificación de reunión

En esta pantalla se muestra un listado con los usuarios que pueden asistir a la reunión. Esta pantalla se mostrará pulsando el botón editar una vez se haya enviado la convocatoria. En ella podremos definir y editar los acuerdos alcanzados, así como los comentarios y observaciones que de ella sucedan.

En clave de rock

**Datos de la reunión**

<b>Fecha de la reunión</b>	27/07/2012	<b>Hora de la reunión</b>	17:30
<b>Lugar de la reunión</b>	Distrito norte		

**Orden del día**

- Estado de los materiales
- Vigilancia a la entrada
- Pérdida de llaves y cambio de cerradura si procede
- Propuestas para la mejora de la seguridad
- Ruegos y preguntas

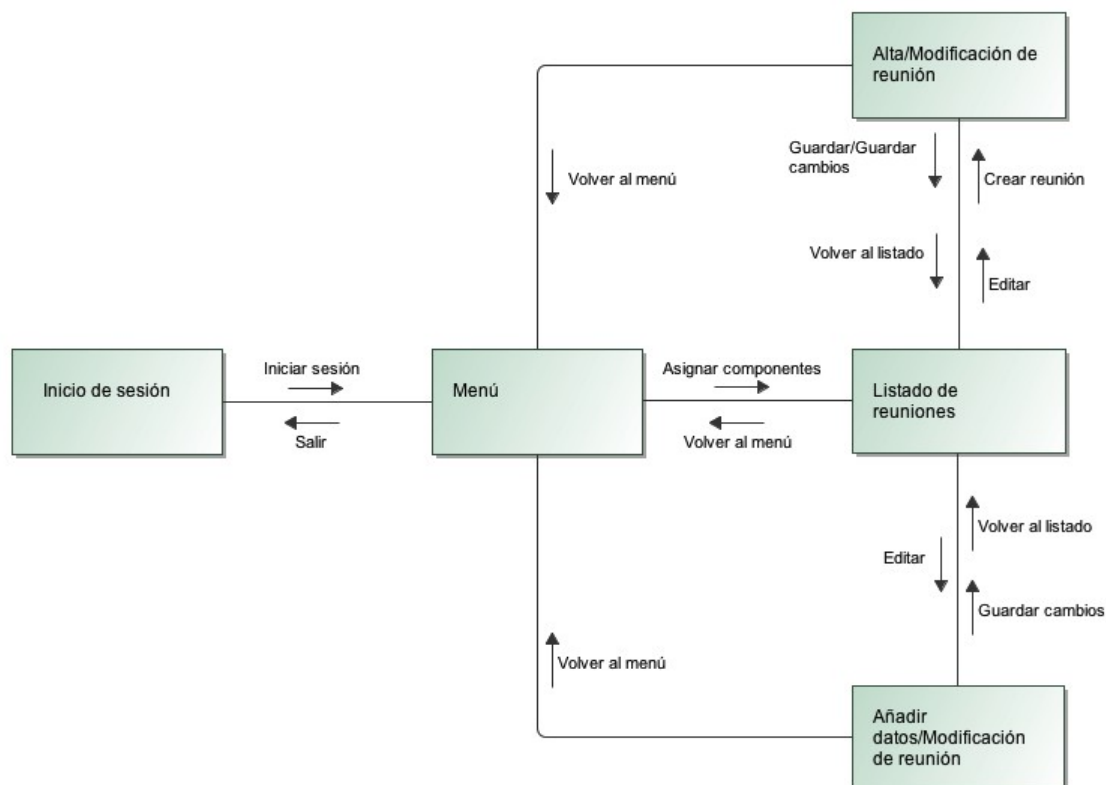
**Acuerdos**

**Observaciones y comentarios**

**Asistentes**

NOMBRE	DNI	EMAIL	ACCIONES
Robert Flynn	65482487F	robbflynn@machinehead1.com	<input type="checkbox"/>
Phill Demmel	68789458D	philldemmel@machinehead1.com	<input type="checkbox"/>
Adam Duce	67546821D	adamduce@machinehead1.com	<input type="checkbox"/>
Dave McClain	67546532M	davemcclain@machinehead1.com	<input type="checkbox"/>

## Diagrama de flujo entre pantallas



## Módulo de solicitudes

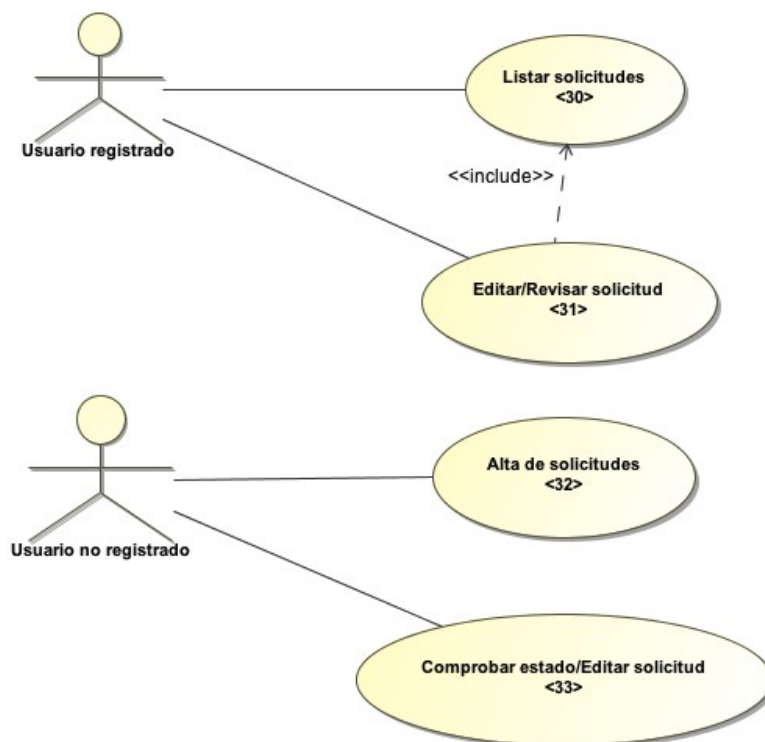
Este módulo permite la realización de solicitudes de acceso a la asociación. Aquí, un usuario externo puede realizar una petición de acceso a la asociación indicando ciertos datos. Se podrá también consultar el estado de la solicitud para ver si el usuario finalmente es admitido en la asociación.

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Sólo los usuarios no registrados pueden crear solicitudes de acceso
- Las solicitudes deberán tener unos datos obligatorios de contacto del solicitante y un campo abierto
- Las solicitudes tienen estado "Pendiente", "Aceptada" o "Rechazada" (sólo uno al mismo tiempo)
- Las solicitudes no se cerrarán y podrán pasar de estado "Aceptada" a "Rechazada" y viceversa
- Sólo los usuarios registrados podrán cambiar el estado de la solicitud
- Cualquier usuario registrado con los permisos necesarios podrá manipular cualquier solicitud



## Casos de uso



### Listar solicitudes

Código	<30>
Resumen	Se muestra un listado con las solicitudes que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar el registro de las solicitudes que existen en la base de datos.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú “Ver solicitudes”</li> <li>3. Se muestra el listado de solicitudes</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> </ul>

### Editar/Revisar solicitud

Código	<31>
Resumen	Se muestra un listado con las solicitudes que almacena la base de datos.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar los datos o revisar una solicitud.
Actores	Usuario registrado
Relacionados	Listar solicitudes <30>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;30&gt;</li> </ol>

Flujo alternativo/excepciones	<ol style="list-style-type: none"> <li>2. Se pulsa el botón "Editar" de la solicitud deseada</li> <li>3. Se revisan los datos, se modifican o se altera el estado de la solicitud y se pulsa el botón "Guardar cambios"</li> <li>4. Los cambios se guardan en el sistema</li> </ol> <ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El listado está vacío</li> <li>• Faltan datos por rellenar</li> </ul>
-------------------------------	--

Alta de solicitud	
Código	<32>
Resumen	Damos de alta una solicitud en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder enviar una solicitud de acceso al portal y registrarla en el sistema.
Actores	Usuario no registrado
Relacionados	
Flujo normal	<ol style="list-style-type: none"> <li>1. Entramos en la pantalla de inicio de sesión</li> <li>2. Se pulsa el botón "Solicitar acceso al portal"</li> <li>3. Se insertan los datos obligatorios y pulsamos el botón "Enviar solicitud"</li> <li>4. La solicitud queda almacenada en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• Faltan datos por rellenar</li> </ul>

Comprobar estado solicitud	
Código	<33>
Resumen	Se comprueba el estado de una solicitud o se modifica.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar el estado o editar los datos de una solicitud previamente enviada.
Actores	Usuario no registrado
Relacionados	
Flujo normal	<ol style="list-style-type: none"> <li>1. Entramos en la pantalla de inicio de sesión</li> <li>2. Se pulsa el botón "Solicitar acceso al portal"</li> <li>3. Se pulsa el botón "Comprobar estado solicitud"</li> <li>4. Vemos el estado de la solicitud</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• La solicitud no existe</li> </ul>

## Prototipos de interfaz gráfica

### Listado de solicitudes

Esta pantalla muestra el listado de solicitudes almacenadas en la base de datos. Se mostrará un pequeño resumen con la fecha y otros datos de la solicitud. Desde aquí podemos volver al menú principal.



### Revisión de solicitud

En esta pantalla, el usuario registrado podrá actualizar el estado de la solicitud y añadir los comentarios que considere oportunos.



### Alta de solicitud

En esta pantalla, el usuario registrado podrá enviar una solicitud de acceso al portal después de haber rellenado algunos datos obligatorios. Desde esta pantalla se podrá acceder a la pantalla para comprobar el estado de la solicitud.

The screenshot shows a window titled "En clave de rock" with a standard Mac OS-style title bar. The main content area is a form with two sections. The first section, "Datos del solicitante", contains six input fields: "Nombre", "Primer apellido", "Segundo apellido", "Teléfono fijo", "Teléfono móvil", and "Correo electrónico", plus a dropdown menu for "Grupo". The second section, "Comentarios", is a large text area. At the bottom of the window, there are three buttons: "Enviar solicitud", "Comprobar estado solicitud", and "Volver al inicio".

### Comprobar estado solicitud

El usuario registrado podrá comprobar, a través de esta pantalla, el estado de la solicitud que previamente haya enviado. En ella se reflejará el veredicto (estado) con los comentarios correspondientes.

En clave de rock

---

**Datos del solicitante**

Nombre <input type="text" value="Dave"/>	Primer apellido <input type="text" value="Mustaine"/>
Segundo apellido <input type="text"/>	Teléfono fijo <input type="text"/>
Teléfono móvil <input type="text"/>	Correo electrónico <input type="text" value="dave@megadeth.com"/>
Grupo <input type="text" value="Megadeth"/>	

---

**Comentarios**

Hola!

A mi grupo y a mí nos gustaría entrar a formar parte de la asociación. Mi nombre es Dave, mi grupo se llama Megadeth de estilo Trash Metal y en principio buscamos horas libres en el fin de semana que estén entre las 11 de la mañana y las 7 de la tarde.

Saludos!

---

**Resolución de la solicitud**

**Estado** Aceptada

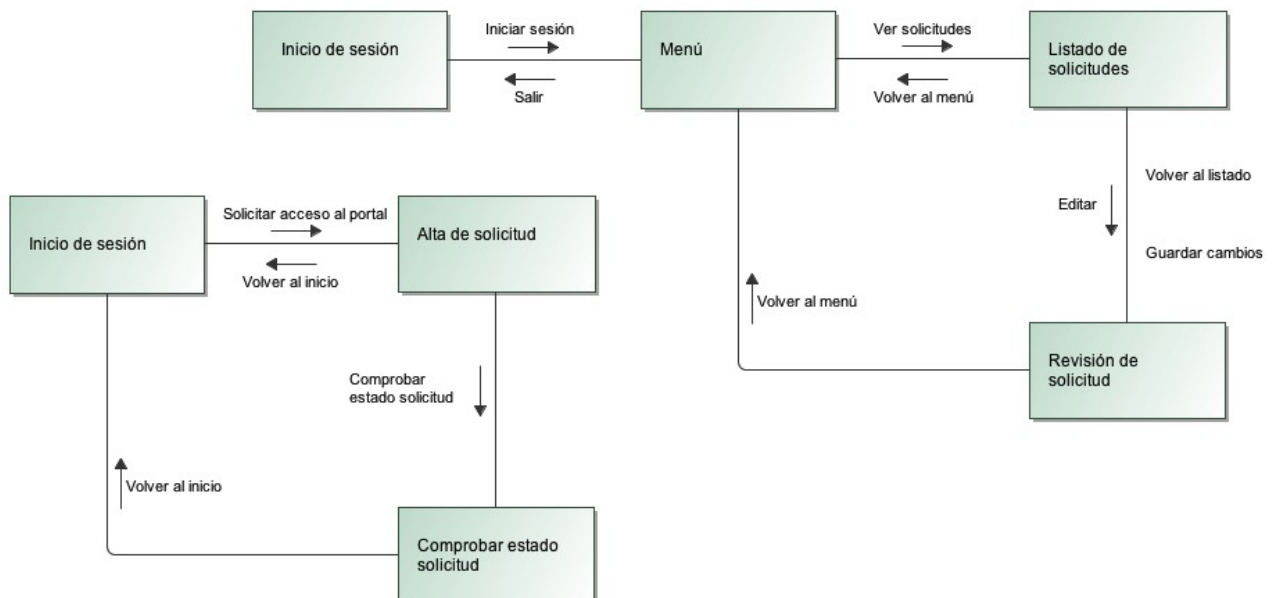
**Comentarios**

Conforme a vuestra solicitud, hemos aceptado asignaros el horario acordado.

¡Bienvenid@s!

---

## Diagrama de flujo entre pantallas



## Módulo de horario

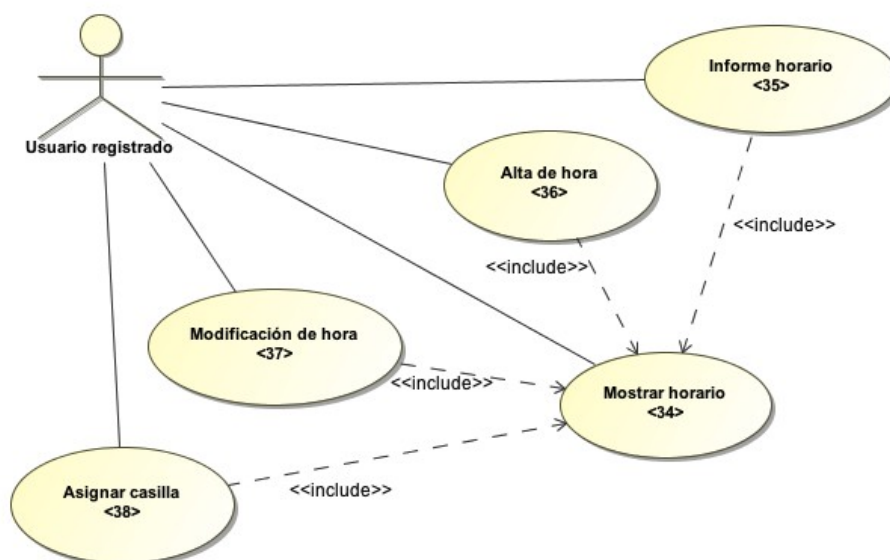
Este módulo permite la gestión de las horas de acceso a la sala de ensayos. En este módulo se podrán modificar, dar de alta

y de baja las horas de ensayo. También se podrán asignar las horas que ocupa cada grupo en el horario o bien quitar al grupo o grupos esas horas. Aquí se permitirá también la generación del esquema con el horario actual de la asociación.

La aplicación debe prestar funcionalidad a los siguientes supuestos:

- Cada grupo podrá tener  $n$  casillas
- Las horas afectan a la semana completa, pero la casilla afectará a un único día de la semana y una única hora
- Existirán unas horas predefinidas que se podrán modificar
- Se podrán crear nuevas horas
- Se podrá obtener un informe en PDF con el cuadrante actual del horario

## Casos de uso



### Mostrar horario

Código	<34>
Resumen	Se muestra un esquema con el horario actual.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder comprobar el esquema con el estado del horario actual almacenado en el sistema.
Actores	Usuario registrado
Relacionados	Inicio de sesión <01>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;01&gt;</li> <li>2. Se accede a la opción de menú "Definir horario"</li> <li>3. Se muestra el esquema actual del horario</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El horario está vacío</li> </ul>

### Informe horario

Código	<35>
Resumen	Obtenemos un informe en PDF del horario.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder obtener el informe con el esquema actual del horario almacenado en el sistema.
Actores	Usuario registrado
Relacionados	Mostrar horario <34>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;34&gt;</li> <li>2. Se pulsa el botón "Exportar en PDF"</li> <li>3. Obtenemos el horario en PDF</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• El horario está vacío</li> <li>• El servidor de conversión no responde</li> </ul>

### Alta de hora

Código	<36>
Resumen	Insertamos un nuevo rango de horas en el sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder insertar un nuevo rango de horas (hora de entrada – hora de salida) en el sistema.
Actores	Usuario registrado
Relacionados	Mostrar horario <34>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;34&gt;</li> <li>2. Se pulsa el botón "Nueva hora"</li> <li>3. Insertamos la hora de entrada y salida y pulsamos el botón "Guardar"</li> <li>4. La hora queda almacenada en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> </ul>

### Modificación de hora

Código	<37>
Resumen	Modificamos un rango de horas del sistema.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder modificar un rango de horas previamente almacenado en el sistema.
Actores	Usuario registrado
Relacionados	Mostrar horario <34>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;34&gt;</li> <li>2. Se pulsa el botón "Editar" del rango deseado</li> <li>3. Modificamos la hora de entrada y salida y pulsamos el botón "Guardar cambios"</li> <li>4. La hora queda almacenada en el sistema</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> <li>• Faltan datos por rellenar</li> </ul>

### Asignar casilla

Código	<38>
Resumen	Asignamos una casilla a un grupo.
Papel en el trabajo del usuario	Es el caso de uso que el usuario ha de seguir para poder asignar un rango de horas en un día de la semana a un grupo determinado.
Actores	Usuario registrado
Relacionados	Mostrar horario <34>
Flujo normal	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso &lt;34&gt;</li> <li>2. Se pulsa en la casilla deseada</li> <li>3. Seleccionamos el grupo y pulsamos "Asignar"</li> <li>4. El grupo queda asignado a la casilla</li> </ol>
Flujo alternativo/excepciones	<ul style="list-style-type: none"> <li>• El usuario no tiene permisos para realizar esta acción</li> </ul>

## Prototipos de interfaz gráfica

### Horario

En esta pantalla veremos un esquema actual con los datos del horario. Desde aquí podremos modificar las casillas, editar los rangos de horas, asignar grupos a las casillas, dar de alta nuevos rangos y generar un informe con el horario actual.

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO	ACCIONES
08:00 - 10:00						Machine Head		Editar
10:00 - 12:00	Coibox		Amaranthe			Machine Head		Editar
12:00 - 14:00	KoRn	KoRn		KoRn				Editar
15:00 - 17:00		KoRn	Amaranthe	KoRn	Machine Head	Coilbox	KoRn	Editar
17:00 - 19:00			Amaranthe		Machine Head	Coilbox	Amaranthe	Editar
19:00 - 21:00					Machine Head			Editar

Nueva hora    Exportar en PDF    Volver al menú

### Alta de hora

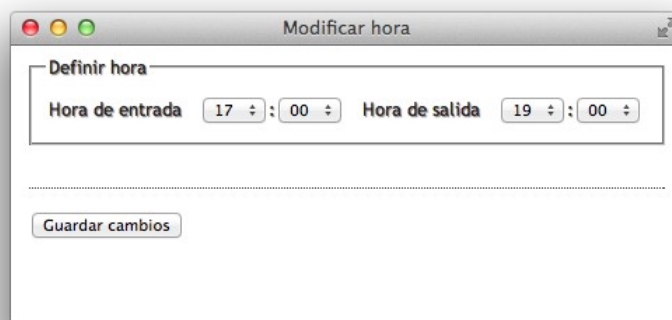
Desde esta pantalla podremos dar de alta un nuevo rango de horas para todos los días de la semana.





### Editar hora

Desde aquí podemos editar un rango de horas que previamente esté registrado en el sistema. Se deberá indicar la hora y el minuto de entrada, así como la hora y el minuto de salida.

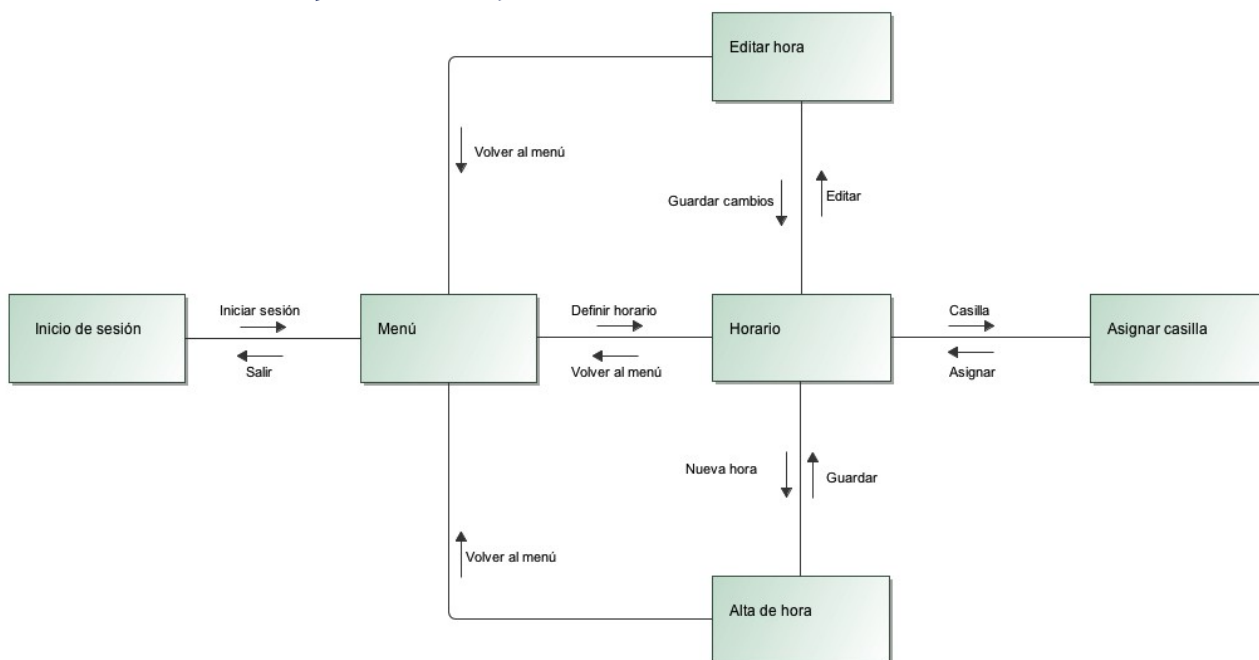


### Asignar casilla

Desde esta pantalla podremos definir el grupo que quedará asignado a una determinada hora en un determinado día de la semana. Antes, deberemos haber seleccionado la casilla del horario y entonces seleccionamos el grupo que pasará a ocupar dicha casilla.



## Diagrama de flujo entre pantallas



## Diseño técnico

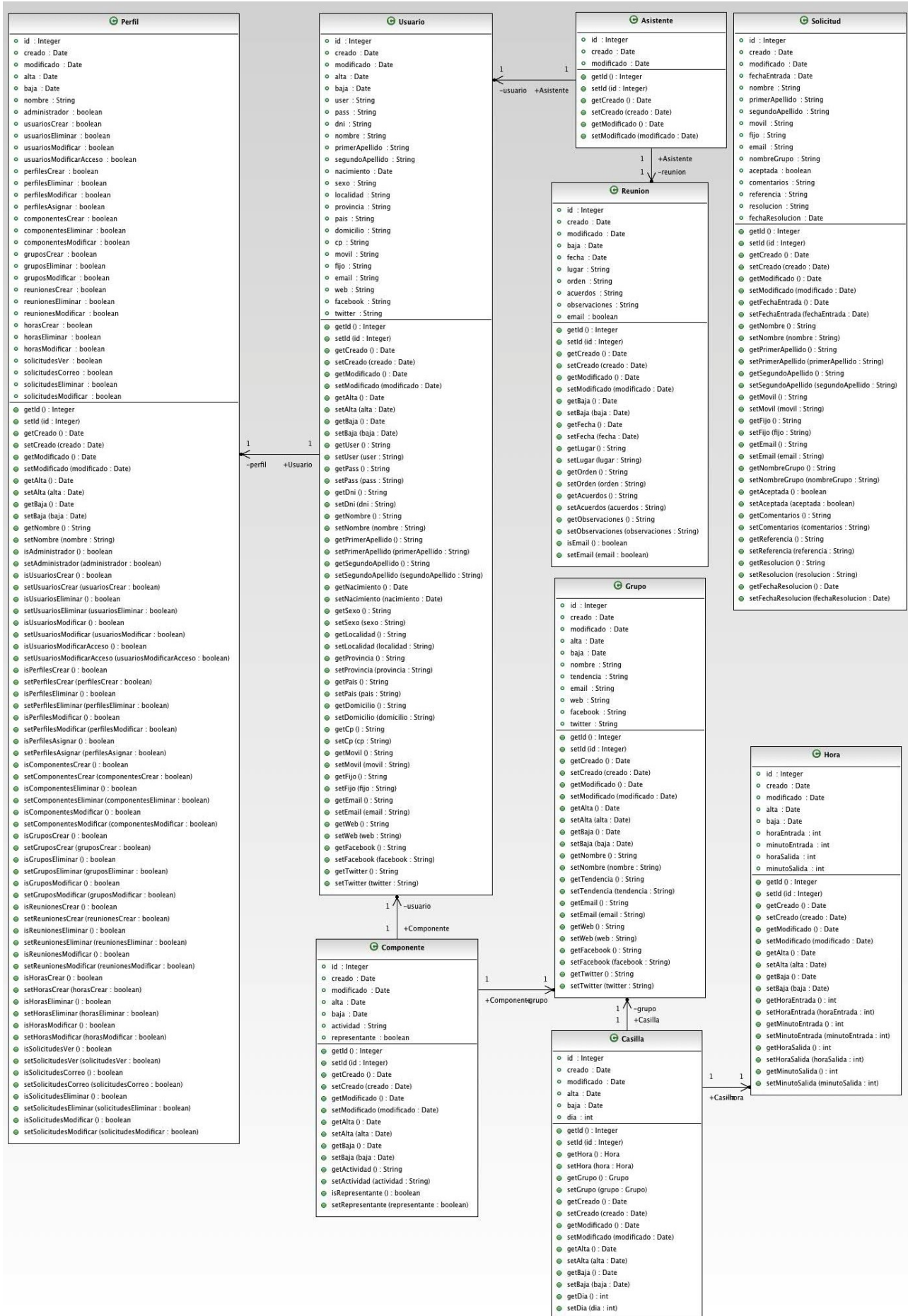
### Diseño de las clases

Las clases estarán divididas en paquetes según su función. Básicamente distinguiremos 3 grandes bloques: Persistencia, Negocio y Utilidades.

#### Persistencia

Dentro de este primer bloque nos encontramos con las clases de **dominio** y los **servicios**. Las clases de dominio intentan representar con atributos Java las columnas de la tabla correspondiente de base de datos. Por tanto, tendremos el mismo número de clases de dominio que tablas en la base de datos. Las clases de servicio englobarán los accesos a base de datos y serán las que actúen con el framework de persistencia (Hibernate) para recuperar, actualizar, crear o eliminar registros.

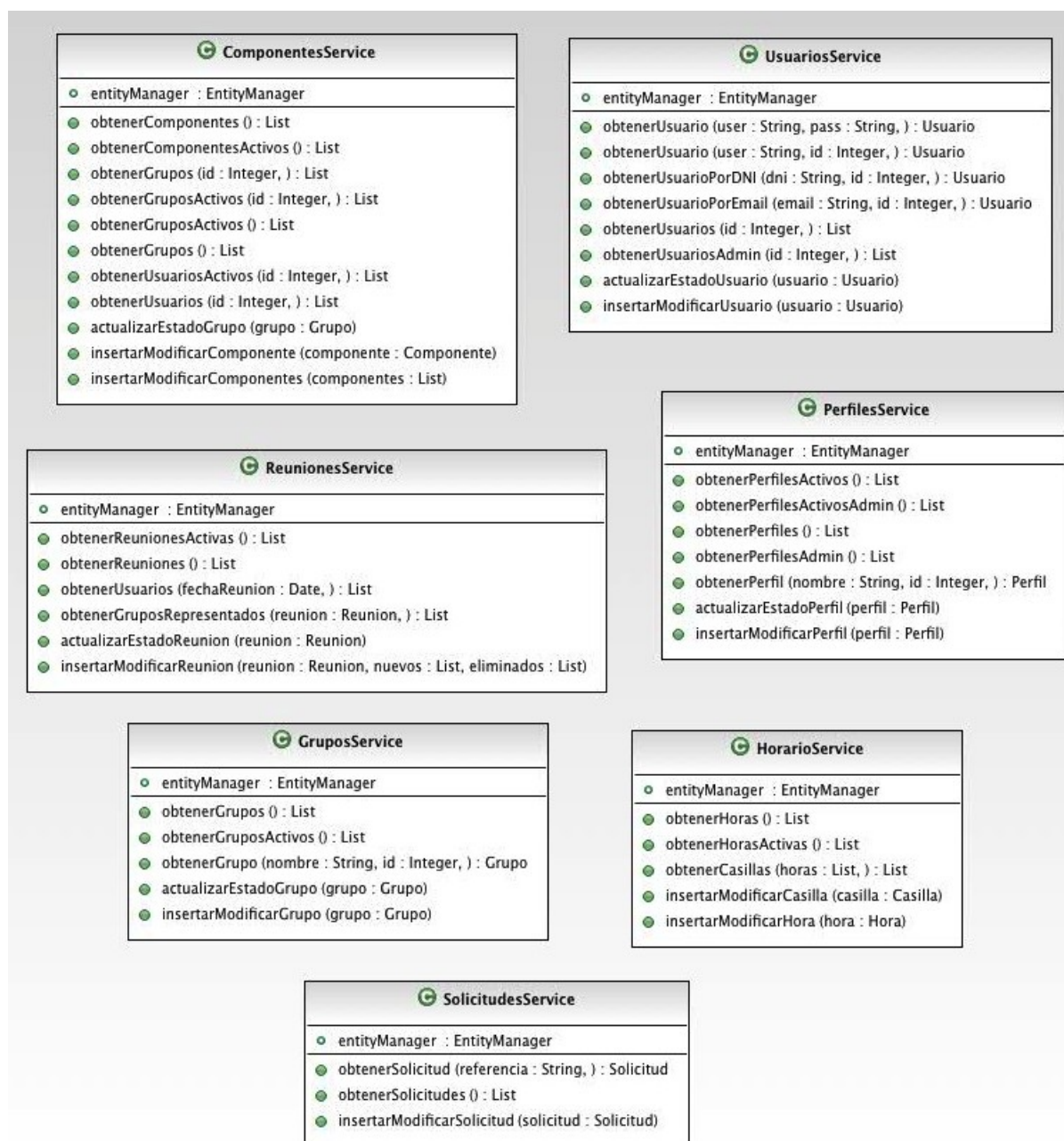
# Dominio



## Servicios

Tendremos un servicio por cada módulo. Cada servicio se encargará de las operaciones con la base de datos relacionadas con las tablas de cada módulo. Objeto común en todos los servicios será el gestor de entidades de JPA EntityManager que se usará para la comunicación de los servicios con el framework de persistencia (Hibernate).

Con los servicios podremos crear, recuperar, actualizar y eliminar registros (CRUD) de las tablas afectadas por cada módulo, esto es, no se harán operaciones sobre la tabla de componentes en el módulo de horario, por ejemplo.



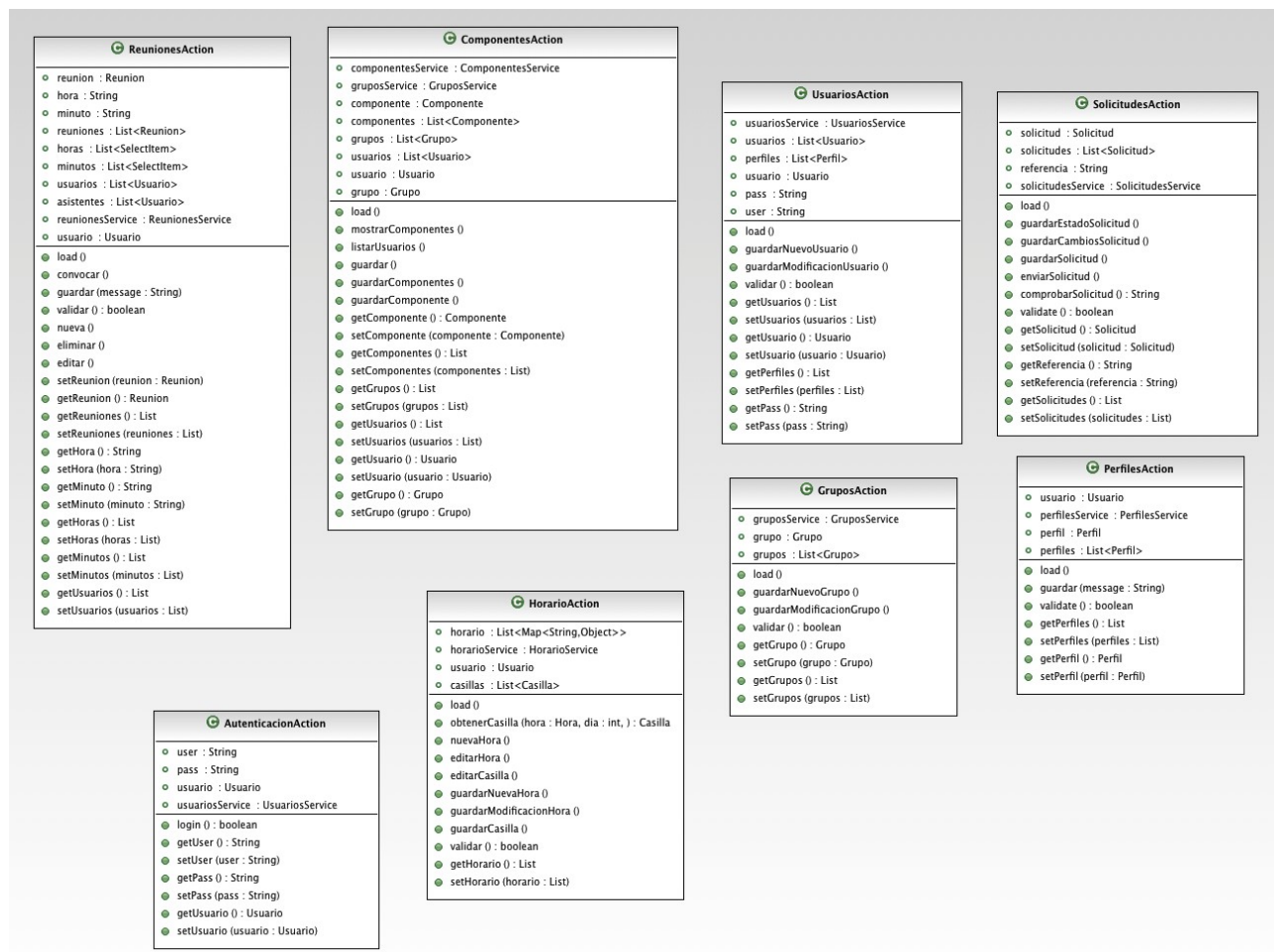
## Negocio

La capa de negocio tiene dos áreas distinguidas: los **manejadores** y los **gestores de informe**. Los manejadores se encargarán de procesar las peticiones que realice el usuario y se encargarán de preparar los datos para la respuesta. Por otro lado, los gestores de informe son un grupo de manejadores encargados de las tareas específicas relacionadas con los

informes.

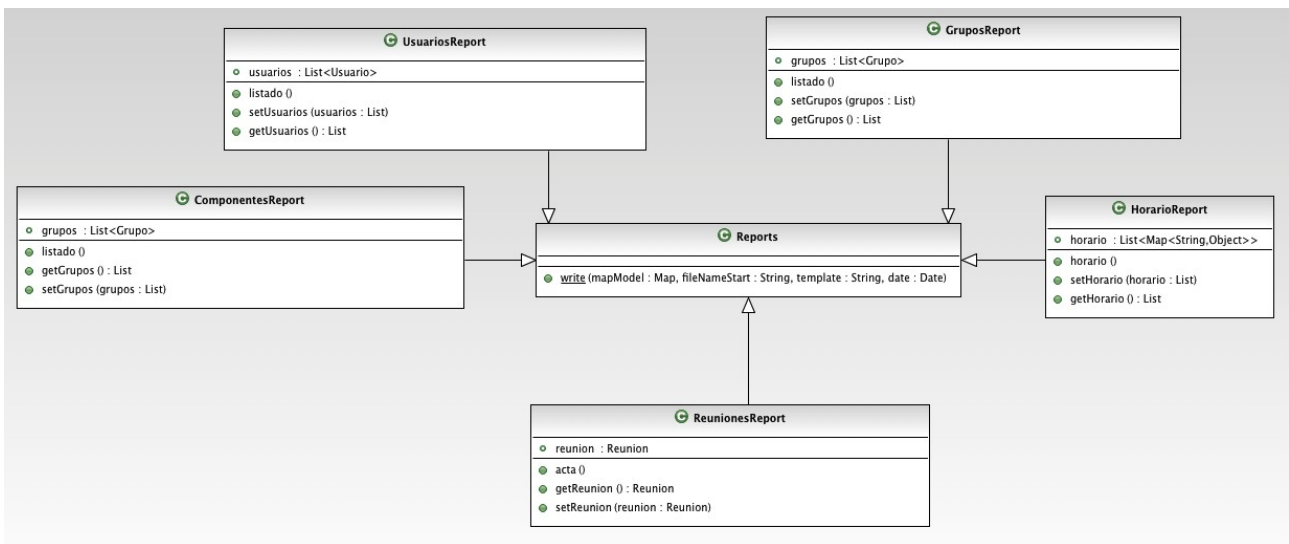
## Manejadores

Existirán por cada módulo o acción concreta que se tenga que realizar. Tendrán métodos de carga inicial que servirán para preparar los datos antes de presentarlos en las vistas.



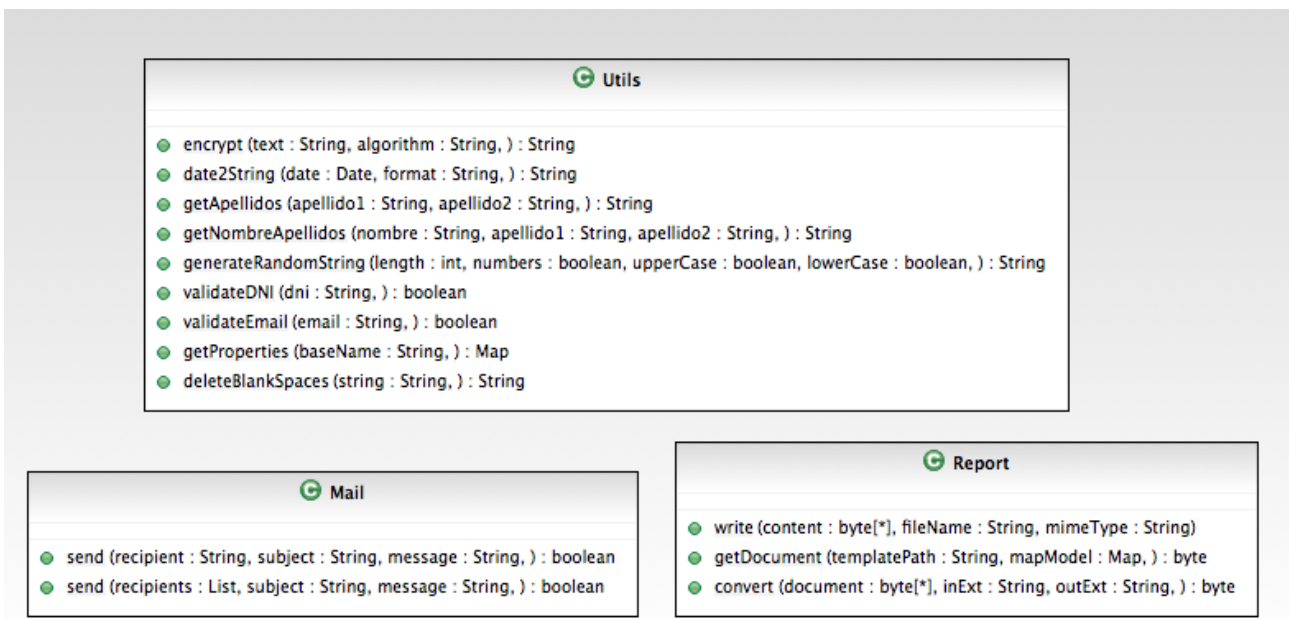
## Gestores de informe

Estas clases serán las encargadas de procesar las peticiones relacionadas con la documentación. Leerán los datos de las plantillas que posteriormente serán enviadas al servidor LibreOffice para su conversión a formato PDF. Tendremos una clase padre que contendrá los métodos que vayan a ser usados por todas las clases hijas de forma que éstas últimas hereden dichos métodos.



## Utilidades

Las clases de utilidades contendrán métodos que nos permitan realizar ciertas tareas repetitivas y aisladas de forma rápida y fácil. Las tareas de conversión de campos, de validación, de formateo, etc. son tareas que solemos usar con frecuencia y que estarán englobadas en las clases de utilidades.



## Detalle de clases

En este punto pasamos a detallar las clases que usaremos en el proyecto. No hemos incluido el paquete de dominios que quedará reflejado en el diseño de la persistencia como el diseño de las tablas.

ComponentesService	
Atributos	Métodos
entityManager:EntityManager	obtenerComponentes():List
	obtenerComponentesActivos():List

obtenerGrupos():List  
 obtenerGrupos(id:Integer):List  
 obtenerGruposActivos():List  
 obtenerGruposActivos(id:Integer):List  
 obtenerUsuarios(id:Integer):List  
 obtenerUsuariosActivos(id:Integer):List  
 actualizarEstadoGrupo(grupo:Grupo)  
 insertarModificarComponente(componente:Componente)  
 insertarModificarComponentes(componentes:List)

### UsuariosService

#### Atributos

entityManager:EntityManager

#### Métodos

obtenerUsuario(user:String, pass:String):Usuario  
 obtenerUsuario(user:String, id:Integer):Usuario  
 obtenerUsuarioPorDNI(dni:String, id:Integer):Usuario  
 obtenerUsuarioPorEmail(email:String, id:Integer):Usuario  
 obtenerUsuarios(id:Integer):List  
 obtenerUsuariosAdmin(id:Integer):List  
 actualizarEstadoUsuario(usuario:Usuario)  
 insertarModificarUsuario(usuario:Usuario)

### ReunionesService

#### Atributos

entityManager:EntityManager

#### Métodos

obtenerReuniones():List  
 obtenerReunionesActivas():List  
 obtenerUsuarios(fechaReunion:Date):List  
 obtenerGruposRepresentados(reunion:Reunion):List  
 actualizarEstadoReunion(reunion:Reunion)  
 insertarModificarReunion(reunion:Reunion, nuevos:List, eliminados:List)

### PerfilesService

#### Atributos

entityManager:EntityManager

#### Métodos

obtenerPerfiles():List  
 obtenerPerfilesActivos():List  
 obtenerPerfilesAdmin():List  
 obtenerPerfilesActivosAdmin():List  
 obtenerPerfil(nombre:String, id:Integer):Perfil

actualizarEstadoPerfil(perfil:Perfil)  
 insertarModificarPerfil(perfil:Perfil)

GruposService	
Atributos	Métodos
entityManager:EntityManager	obtenerGrupos():List obtenerGruposActivos():List obtenerGrupo(nombre:String, id:Integer):Grupo actualizarEstadoGrupo(grupo:Grupo) insertarModificarGrupo(grupo:Grupo)

HorarioService	
Atributos	Métodos
entityManager:EntityManager	obtenerHoras():List obtenerHorasActivas():List obtenerCasillas(horas:List):List obtenerUsuarioPorEmail(email:String, id:Integer):Usuario insertarModificarCasilla(casilla:Casilla) insertarModificarHora(hora:Hora)

SolicitudesService	
Atributos	Métodos
entityManager:EntityManager	obtenerSolicitud(referencia:String):Solicitud obtenerSolicitudes():List insertarModificarSolicitud(solicitud:Solicitud)

ReunionesAction	
Atributos	Métodos
reunion:Reunion	load()
hora:String	convocar()
minuto:String	guardar(message:String)
reuniones:List	validar():boolean
horas:List	nueva()
minutos:List	eliminar()
usuarios:List	editar()
asistentes:List	setReunion(reunion:Reunion)
reunionesService:ReunionesService	getReunion():Reunion



usuario:Usuario

setReuniones(reuniones:List)  
 getReuniones():List  
 setHora(hora:String)  
 getHora():String  
 setMinuto(minuto:String)  
 getMinuto():String  
 setHoras(horas:List)  
 getHoras():List  
 setMinutos(minutos:List)  
 getMinutos():List  
 setUsuarios(usuarios:List)  
 getUsuarios():List

### ComponentesAction

Atributos

Métodos

componentesService:ComponentesService  
 gruposService:GruposService  
 componente:Componente  
 componentes:List  
 grupos:List  
 usuarios:List  
 usuario:Usuario  
 grupo:Grupo

load()  
 mostrarComponentes()  
 listarUsuarios()  
 guardar()  
 guardarComponentes()  
 guardarComponente()  
 setComponentes(componentes:List)  
 getComponentes():List  
 setGrupos(grupos:List)  
 getGrupos():List  
 setUsuarios(usuarios:List)  
 getUsuarios():List  
 setGrupo(grupo:Grupo)  
 getGrupo():Grupo

### UsuariosAction

Atributos

Métodos

usuariosService:UsuariosService  
 usuarios:List  
 perfiles:List  
 usuario:Usuario  
 pass:String

load()  
 guardarNuevoUsuario()  
 guardarModificacionUsuario()  
 validar()  
 getUsuarios():List

user:String

setUsuarios(usuarios:List)  
 getUsuario():Usuario  
 setUsuario(usuario:Usuario)  
 getPerfiles():List  
 setPerfiles(perfiles:List)  
 getPass():String  
 setPass(pass:String)  
 getUser():String  
 setUser(user:String)

### SolicitudesAction

Atributos

Métodos

solicitud:Solicitud  
 solicitudes:List  
 referencia:String  
 solicitudesService:SolicitudesService

load()  
 guardarEstadoSolicitud()  
 guardarCambiosSolicitud()  
 guardarSolicitud()  
 enviarSolicitud()  
 comprobarSolicitud():String  
 validate():boolean  
 getSolicitudes():List  
 setSolicitudes(solicitudes:List)  
 getReferencia():String  
 setReferencia(referencia:String)  
 getSolicitudes():List  
 setSolicitudes(solicitudes:List)

### PerfilesAction

Atributos

Métodos

usuario:Usuario  
 perfilesService:PerfilesService  
 perfil:Perfil  
 perfiles:List

load()  
 guardar(message:String)  
 validate():boolean  
 getPerfiles():List  
 setPerfiles(perfiles:List)  
 getPerfil():Perfil  
 setPerfil(perfil:Perfil)

### GruposAction

Atributos	Métodos
gruposService:GruposService	load()
grupo:Grupo	guardarNuevoGrupo()
grupos:List	guardarModificacionGrupo()
	validar()
	getGrupo():Grupo
	setGrupo(grupo:Grupo)
	getGrupos():List
	setGrupos(grupos:List)

### HorarioAction

Atributos	Métodos
horario:List	load()
horarioService:HorarioService	obtenerCasilla(hora:Hora, dia:int):Casilla
usuario:Usuario	nuevaHora()
casillas:List	editarHora()
	editarCasilla()
	guardarNuevaHora()
	guardarModificacionHora()
	guardarCasilla()
	validar()
	getHorario():List
	setHorario(horario:List)

### AutenticacionAction

Atributos	Métodos
user:String	login()
pass:String	getUser():String
usuario:Usuario	setUser(user:String)
usuariosService:UsuariosService	getPass():String
	setPass(pass:String)
	getUsuario():Usuario
	setUsuario(usuario:Usuario)

### Reports

Atributos	Métodos
	write(mapModel:Map, fileNameStart:String, template:String, date:Date)

## UsuariosReport

### Atributos

usuarios:List

### Métodos

listado()  
 setUsuarios(usuarios:List)  
 getUsuarios():List

## GruposReport

### Atributos

grupos:List

### Métodos

listado()  
 setGrupos(grupos:List)  
 getGrupos():List

## ComponentesReport

### Atributos

grupos:List

### Métodos

listado()  
 setGrupos(grupos:List)  
 getGrupos():List

## HorarioReport

### Atributos

horario:List

### Métodos

horario()  
 setHorario(horario:List)  
 getHorario():List

## ReunionesReport

### Atributos

reunion:Reunion

### Métodos

acta()  
 setReunion(reunion:Reunion)  
 getReunion():Reunion

## Utils

### Atributos

### Métodos

encrypt(text:String, algorithm:String):String  
 date2String(date:Date, format:String):String  
 getApellidos(apellido1:String, apellido2:String):String  
 getNombreApellidos(nombre:String, apellido1:String, apellido2:String):String

```
generateRandomString(length:int, numbers:boolean, upperCase:boolean,
lowerCase:boolean):String
validateDNI(dni:String):boolean
getProperties(baseName:String):Map
deleteBlankSpaces(string:String):String
```

Mail	
Atributos	Métodos

```
send(recipient:String, subject:String, message:String):boolean
send(recipients:List, subject:String, message:String):boolean
```

Report	
Atributos	Métodos

```
write(content:byte[], fileName:String, mimeType:String)
getDocument(templatePath:String, mapModel:Map): byte[]
convert(document:byte[], inExt:String, outExt:String):byte[]
```

## Diseño de la persistencia

En este apartado vamos a exponer el diseño conceptual de las tablas y el diagrama Entidad/Relación del modelo de datos relacional.

### Diseño de las tablas

**grupos** (id, creado, modificado, alta, baja, nombre, tendencia, email, web, facebook, twitter)

**horas** (id, creado, modificado, alta, baja, hora\_entrada, minuto\_entrada, hora\_salida, minuto\_salida)

**perfiles** (id, creado, modificado, alta, baja, nombre, administrador, usuarios\_crear, usuarios\_eliminar, usuarios\_modificar, usuarios\_modificar\_acceso, perfiles\_crear, perfiles\_eliminar, perfiles\_modificar, perfiles\_asignar, componentes\_crear, componentes\_eliminar, componentes\_modificar, grupos\_crear, grupos\_eliminar, grupos\_modificar, reuniones\_crear, reuniones\_eliminar, reuniones\_modificar, horas\_crear, horas\_eliminar, horas\_modificar, solicitudes\_ver, solicitudes\_correo, solicitudes\_eliminar, solicitudes\_modificar)

**reuniones** (id, creado, modificado, baja, fecha, lugar, orden, acuerdos, observaciones, email)

**solicitudes** (id, creado, modificado, nombre, primer\_apellido, segundo\_apellido, movil, fijo, email,

nombre\_grupo, aceptada, comentarios, referencia, fecha\_entrada, resolucion, fecha\_resolucion)

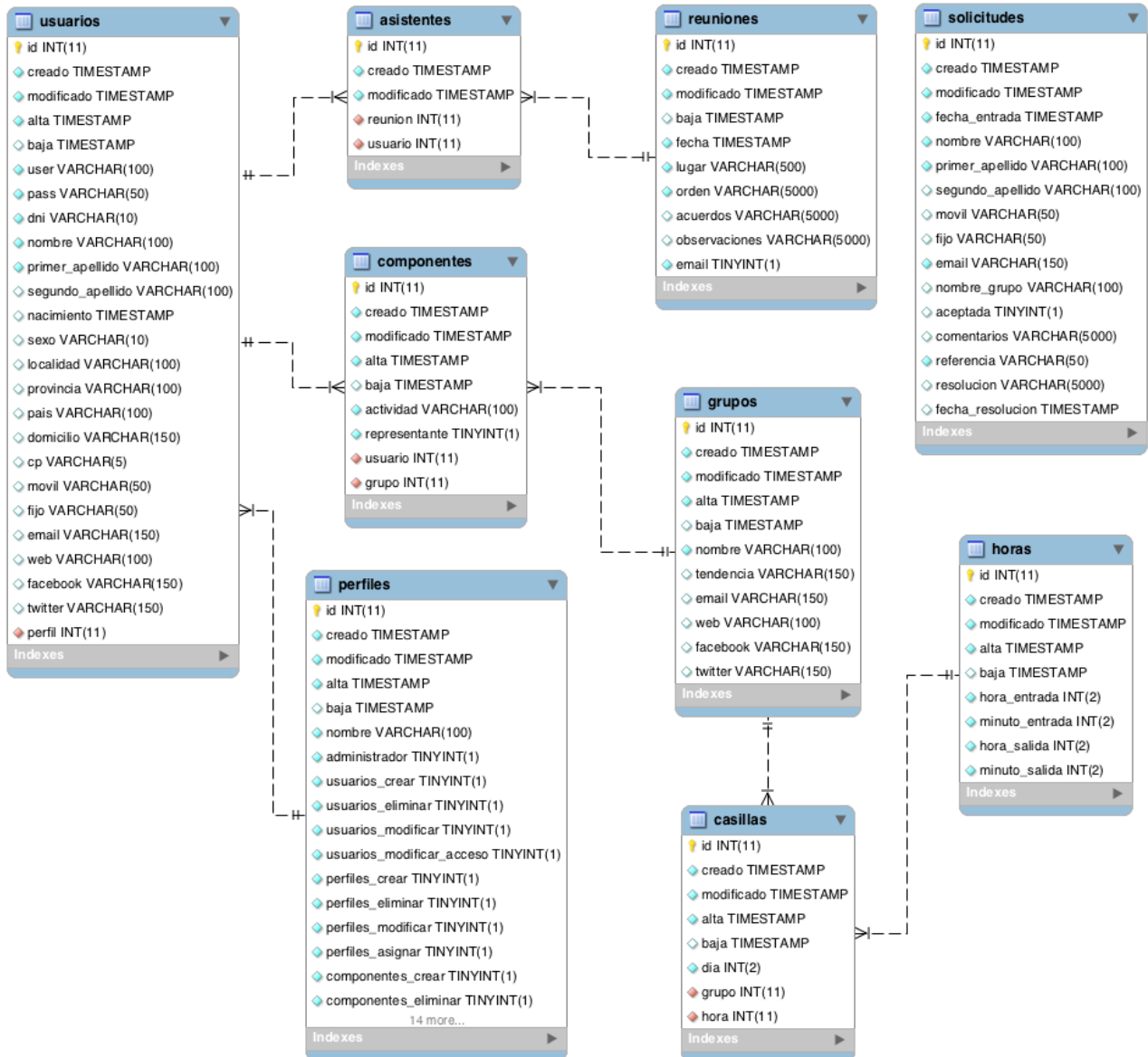
**usuarios** (id, creado, modificado, alta, baja, user, pass, dni, nombre, primer\_apellido, localidad, provincia, pais, domicilio, cp, movil, fijo, email, web, facebook, twitter, nacimiento, sexo, {perfil})  
{perfil} es clave ajena hacia **perfiles**

**asistentes** (id, creado, modificado, {reunion}, {usuario})  
{reunion} es clave ajena hacia **reuniones**  
{usuario} es clave ajena hacia **usuarios**

**casillas** (id, creado, modificado, alta, baja, dia, {grupo}, {hora})  
{grupo} es clave ajena hacia **grupos**  
{hora} es clave ajena hacia **horas**

**componentes** (id, creado, modificado, alta, baja, actividad, representante, {usuario}, {grupo})  
{usuario} es clave ajena hacia **usuarios**  
{grupo} es clave ajena hacia **grupos**

## Diagrama E/R



## Detalle de tablas

Grupos						
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros	
id	int(11)	No	Primaria		auto_increment	
creado	timestamp	No				
modificado	timestamp	No				
alta	timestamp	No				
baja	timestamp	Sí		null		
nombre	varchar(100)	No	Única			
tendencia	varchar(150)	Sí		null		
email	varchar(150)	Sí		null		

web	varchar(100)	Sí	null
facebook	varchar(150)	Sí	null
twitter	varchar(150)	Sí	null

Horas					
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
alta	timestamp	No			
baja	timestamp	Sí		null	
hora_entrada	int(2)	No			
minuto_entrada	int(2)	No			
hora_salida	int(2)	No			
minuto_salida	int(2)	No			

Perfiles						
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros	
id	int(11)	No	Primaria			auto_increment
creado	timestamp	No				
modificado	timestamp	No				
alta	timestamp	No				
baja	timestamp	Sí		null		
nombre	varchar(100)	No	Única			
administrador	tinyint(1)	No				
usuarios_crear	tinyint(1)	No		null		
usuarios_eliminar	tinyint(1)	No		null		
usuarios_modificar	tinyint(1)	No		null		
usuarios_modificar_acceso	tinyint(1)	No		null		
perfiles_crear	tinyint(1)	No				
perfiles_eliminar	tinyint(1)	No				
perfiles_modificar	tinyint(1)	No				
perfiles_asignar	tinyint(1)	No				
componentes_crear	tinyint(1)	No				
componentes_eliminar	tinyint(1)	No				
componentes_modificar	tinyint(1)	No				
grupos_crear	tinyint(1)	No				



grupos_eliminar	tinyint(1)	No
grupos_modificar	tinyint(1)	No
reuniones_crear	tinyint(1)	No
reuniones_eliminar	tinyint(1)	No
reuniones_modificar	tinyint(1)	No
horas_crear	tinyint(1)	No
horas_eliminar	tinyint(1)	No
horas_modificar	tinyint(1)	No
solicitudes_ver	tinyint(1)	No
solicitudes_correo	tinyint(1)	No
solicitudes_eliminar	tinyint(1)	No
solicitudes_modificar	tinyint(1)	No

Reuniones					
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
baja	timestamp	Sí		null	
fecha	timestamp	No			
lugar	varchar(500)	No			
orden	varchar(5000)	No			
acuerdos	varchar(5000)	Sí		null	
observaciones	varchar(5000)	Sí		null	
email	tinyint(1)	No			

Solicitudes					
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
fecha_entrada	timestamp	No			
nombre	varchar(100)	No			
primer_apellido	varchar(100)	No			
segundo_apellido	varchar(100)	Sí		null	
movil	varchar(50)	Sí		null	
fijo	varchar(50)	Sí		null	

email	varchar(150)	No	
nombre_grupo	varchar(100)	Sí	null
aceptada	tinyint(1)	Sí	null
comentarios	varchar(5000)	Sí	null
referencia	varchar(50)	No	
resolucion	varchar(5000)	Sí	null
fecha_resolucion	timestamp	Sí	null

Usuarios					
Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
alta	timestamp	No			
baja	timestamp	Sí		null	
user	varchar(100)	No			
pass	varchar(50)	No			
dni	varchar(10)	No	Única		
nombre	varchar(100)	No			
primer_apellido	varchar(100)	No			
segundo_apellido	varchar(100)	Sí		null	
nacimiento	timestamp	Sí		null	
sexo	varchar(10)	Sí		null	
localidad	varchar(100)	Sí		null	
provincia	varchar(100)	Sí		null	
pais	varchar(100)	Sí		null	
domicilio	varchar(150)	Sí		null	
cp	varchar(5)	Sí		null	
movil	varchar(50)	Sí		null	
fijo	varchar(50)	Sí		null	
email	varchar(150)	No	Única		
web	varchar(100)	Sí		null	
facebook	varchar(150)	Sí		null	
twitter	varchar(150)	Sí		null	
perfil	int(11)	No	Ajena		

Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
reunion	int(11)	No	Ajena		
usuario	int(11)	No	Ajena		

### Casillas

Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
alta	timestamp	No			
baja	timestamp	Sí		null	
dia	int(2)	No			
grupo	int(11)	No	Ajena		
hora	int(11)	No	Ajena		

### Componentes

Campo	Tipo	¿NULL?	Clave	Valor defecto	Otros
id	int(11)	No	Primaria		auto_increment
creado	timestamp	No			
modificado	timestamp	No			
alta	timestamp	No			
baja	timestamp	Sí		null	
actividad	varchar(100)	No			
representante	tinyint(1)	No			
usuario	int(11)	No	Ajena		
grupo	int(11)	No	Ajena		

## Arquitectura

La aplicación estará desarrollada en 5 capas. Estas capas estarán comunicadas de forma bidireccional y cada una podrá enviar peticiones y recibir respuestas.

## Capas

1. **Vistas.** Estará formada por el conjunto de pantallas que permitirán la interacción del usuario con la aplicación y viceversa. El usuario recibirá las respuestas en formato HTML y éste se comunicará con el servidor mediante peticiones HTTP vía navegador web.
2. **Negocio.** La forman un conjunto de manejadores encargados de procesar las peticiones del usuario.
3. **Conversión de documentos.** Está referida al servidor que permite la conversión de los documentos generados por la aplicación en distintos formatos.
4. **Persistencia.** Engloba los dominios y los servicios que permiten el acceso y la gestión de la persistencia de los datos de la aplicación.
5. **Base de datos.** Está formada por el sistema gestor de base de datos encargado del almacenamiento de los datos.

## Tecnología

---

Para la realización del proyecto contaremos con los siguientes elementos:

- Entorno de desarrollo Eclipse 3.6 Java EE
- Apache Tomcat 6.0.29
- SVN proporcionado por Google Code para el versionado del proyecto
- Planner para la realización de la planificación y gestión del proyecto
- MySQL Server 5.1 como sistema gestor de base de datos
- MySQL Workbench 5.2 como herramienta de administración de base de datos y generador de modelo E/R
- MagicDraw 16.5 con licencia de estudiante para la generación de diagramas UML
- LibreOffice.org 3.5 para la generación de la documentación de la aplicación y del proyecto

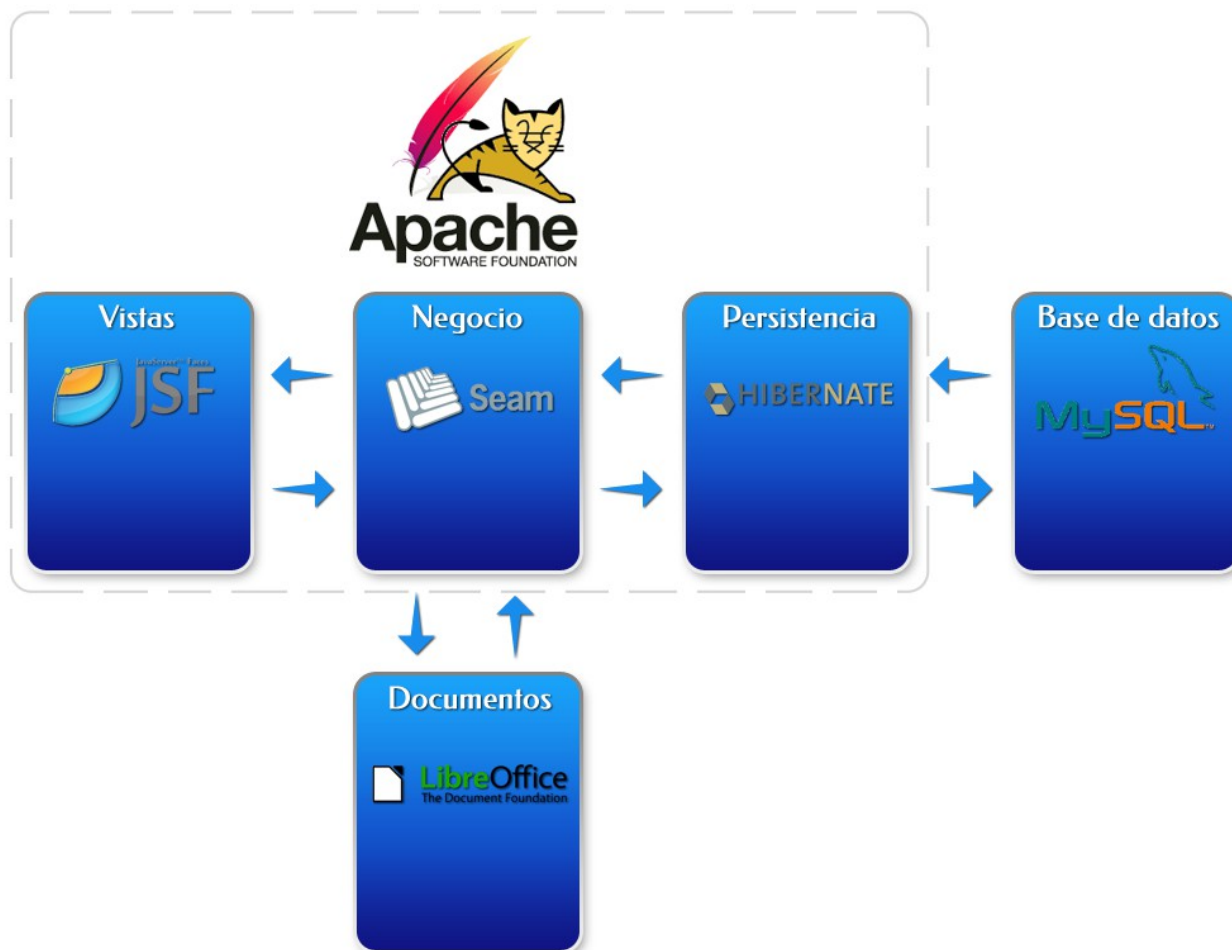
## Frameworks

Esta aplicación contará con algunos de los frameworks más famosos de desarrollo en el mundo J2EE. Éstos se utilizarán tanto para la gestión de peticiones, como para la presentación y la persistencia.

- **JSF.** Es una especificación de Sun Microsystems que pretende dar las pautas para el desarrollo de aplicaciones web. En este proyecto, usaremos además **RichFaces**, que logra una integración perfecta con JSF aportándonos interfaces de usuario ricas en efectos con componentes avanzados.
- **JPA.** Es una especificación de Sun Microsystems que pretende dar las pautas para el desarrollo de modelos de persistencia. En nuestro caso, hacemos uso de **Hibernate**, que hace uso de las anotaciones que presta JPA para las operaciones comunes sobre los sistemas gestores de bases de datos. Además, Hibernate nos permite abstraer la capa de persistencia del SGBD gracias a los dialectos que incorpora el framework. De esta forma, podemos cambiar nuestro SGBD y el dialecto y nuestra aplicación seguiría funcionando sin más complicaciones.

- **Seam.** Es la unión de varias tecnologías (JSF + JPA + EJB). Pretende facilitar el desarrollo de aplicaciones web. Introduce 2 nuevos contextos (*Conversation* y *Business Process*) y permite acceder a cualquier componente accediendo mediante el nombre de Componente Seam asignado.

## Diagrama de componentes



## Planificación

### Hitos y distribución

Tarea	Días	Inicio	Fin
Fase 1. Análisis previo	13	02/03/2012	14/03/2012
Recogida de información del cliente	2	02/03/2012	03/03/2012
Elaboración del análisis y la documentación	6	04/03/2012	09/03/2012
Validación del análisis por el cliente	1	10/03/2012	10/03/2012

Elaboración del plan de trabajo	3	11/03/2012	13/03/2012
Revisión y entrega de la documentación	1	14/03/2012	14/03/2012

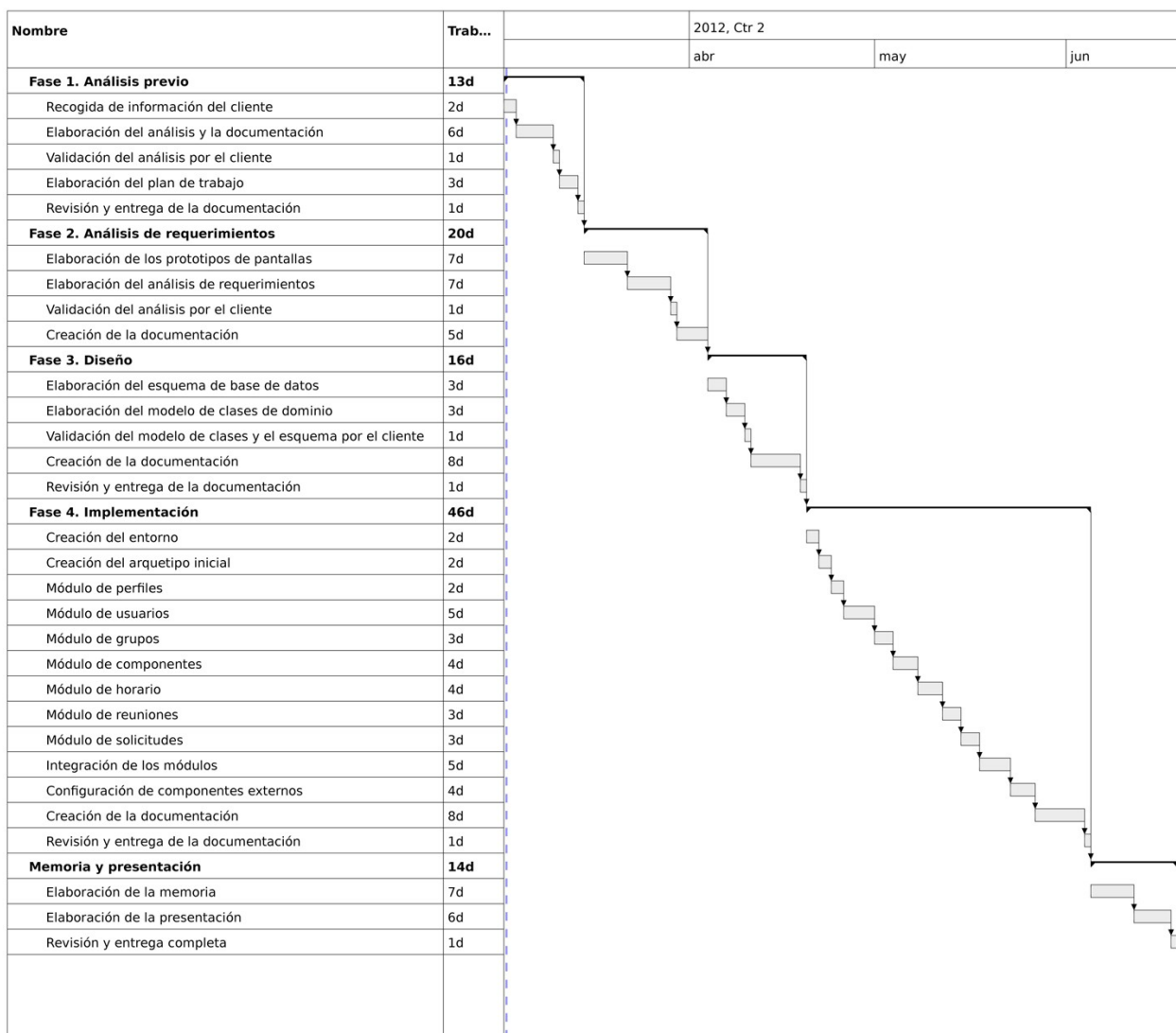
Fase 2. Análisis de requerimientos	20	15/03/2012	03/04/2012
Elaboración de los prototipos de pantallas	7	15/03/2012	21/03/2012
Elaboración del análisis de requerimientos	7	22/03/2012	28/03/2012
Validación del análisis por el cliente	1	29/03/2012	29/03/2012
Creación de la documentación	5	30/03/2012	03/04/2012

Fase 3. Diseño	16	04/04/2012	19/04/2012
Elaboración del esquema de base de datos	3	04/04/2012	06/04/2012
Elaboración del modelo de clases de dominio	3	07/04/2012	09/04/2012
Validación del modelo de clases y el esquema por el cliente	1	10/04/2012	10/04/2012
Creación de la documentación	8	11/04/2012	18/04/2012
Revisión y entrega de la documentación	1	19/04/2012	19/04/2012

Fase 4. Implementación	46	20/04/2012	04/06/2012
Creación del entorno	2	20/04/2012	21/04/2012
Creación del arquetipo inicial	2	22/04/2012	23/04/2012
Módulo de perfiles	2	24/04/2012	25/04/2012
Módulo de usuarios	5	26/04/2012	30/04/2012
Módulo de grupos	3	01/05/2012	03/05/2012
Módulo de componentes	4	04/05/2012	07/05/2012
Módulo de horario	4	08/05/2012	11/05/2012
Módulo de reuniones	3	12/05/2012	14/05/2012
Módulo de solicitudes	3	15/05/2012	17/05/2012
Integración de los módulos	5	18/05/2012	22/05/2012
Configuración de componentes externos	4	23/05/2012	26/05/2012
Creación de la documentación	8	27/05/2012	03/06/2012
Revisión y entrega de la documentación	1	04/06/2012	04/06/2012

Memoria y presentación	14	05/06/2012	18/06/2012
Elaboración de la memoria	7	05/06/2012	11/06/2012
Elaboración de la presentación	6	12/06/2012	17/06/2012
Revisión y entrega completa	1	18/06/2012	18/06/2012

# Diagrama de Gantt



## Implantación

El proyecto tiene la estructura propia de un proyecto basado en Maven. Maven es una herramienta diseñada para la gestión y construcción de proyectos Java.

Una de las grandes utilidades que tiene Maven es la posibilidad de definir filtros (filters) de recursos específicos del entorno donde vayamos a desplegar la aplicación web. Se podrán crear tantos filtros como entornos se tengan (por lo general se tendrán dos entornos: desarrollo y producción) y luego se podrán definir los parámetros en ficheros de configuración (archivos *.properties* en Java) específicos para cada entorno.

## Configuraci3n de los recursos

En nuestro caso hemos definido dos ficheros de configuraci3n para dos entornos: desarrollo y producci3n. Los ficheros se llaman `development.properties` y `production.properties`. Los atributos a especificar son los mismos en estos dos ficheros, pero los valores ser3n distintos para cada entorno. El contenido del fichero es el siguiente:

```
hibernate.connection.url
hibernate.connection.username
hibernate.connection.password

libreoffice.host
libreoffice.port

mail.smtp.host
mail.from.email
mail.smtp.user
mail.smtp.pass

log4j.rootLogger.level
log4j.appender.file.File
log4j.appender.file.MaxFileSize
log4j.appender.file.MaxBackupIndex
log4j.digester.level
log4j.seam.level
log4j.hibernate.level
log4j.beanutils.level
log4j.ajax4jsf.level
log4j.richfaces.level
```

Cada lnea del fichero lleva un comentario asociado que nos ayudar3 a completar el valor de la propiedad. Es importante completar el/los fichero/s antes de seguir adelante, ya que los siguientes pasos son dependientes de esta configuraci3n.

## Generaci3n del archivo WAR

Como ya hemos dicho, el proyecto lleva una estructura Maven. Las librerías de las que depende el proyecto est3n especificadas en el archivo `pom.xml` de cada m3dulo. En el `pom.xml` del proyecto padre (*tf*) est3n declarados los repositorios de librerías de los que se obtendr3n las dependencias del proyecto cuando se genere el archivo WAR. Para ello ser3 necesario tener instalado Maven2 y la JVM 1.6 de Oracle Sun.



Una vez tengamos correctamente instalado Maven y la JVM correspondiente, entraremos en una consola y accederemos al directorio d́nde se encuentra ubicado el archivo `pom.xml` del proyecto padre (*tfc*) y una vez estemos alĺ, dependiendo del entorno, ejecutaremos lo siguiente:

```
mvn clean package -P entorno
```

La palabra **entorno** debeŕ ser sustituida por el perfil que indique el entorno donde vayamos a desplegar la aplicaci3n, es decir, para el entorno de desarrollo tenemos el perfil **development**, por tanto, la ejecuci3n debeŕ ser exactamente la siguiente:

```
mvn clean package -P development
```

En el caso del entorno de producci3n tenemos el perfil **production**, por tanto, la ejecuci3n debeŕ ser exactamente la que sigue a continuaci3n:

```
mvn clean package -P production
```

Una vez que hayamos ejecutado la sentencia, se generaŕ un archivo WAR (*tfc.war*) en la carpeta *target* del m3dulo web (*tfc-web*) con las librerías y la configuraci3n propia del entorno que hayamos definido en el comando.

## Configuraci3n de LibreOffice en modo servidor

La aplicaci3n genera una serie de informes que dependen de un servidor de conversi3n de documentos LibreOffice. Para este proyecto hemos usado la versi3n 3.5 de LibreOffice que podemos descargar de forma gratuita para casi cualquier plataforma en su ṕgina oficial [LibreOffice.org](http://LibreOffice.org).

Para este ejemplo hemos usado un equipo con un sistema operativo basado en Unix. La ejecuci3n de la aplicaci3n para arrancar LibreOffice en modo servidor pasa por situarnos en la carpeta donde tengamos el archivo *soffice* que seŕ el encargado de abrir el socket. Una vez situados en esta carpeta, ejecutamos el siguiente comando:

```
sh soffice --accept="socket,host=0.0.0.0,port=8100;urp;StarOffice.ServiceManager"
```

Con esto estaremos indicando a LibreOffice que debe aceptar peticiones de cualquier direcci3n (**host=0.0.0.0**) para el puerto **8100** (**port=8100**). Hay que recordar que estos parámetros son los que hemos indicado en el fichero de filtros, en concreto, la propiedad `Libreoffice.port`.

## Instalación del driver JDBC de MySQL en Apache Tomcat

Descargamos el archivo ZIP que contiene los fuentes y los binarios del driver JDBC desde la página [web de MySQL](#). Dentro del ZIP encontraremos un archivo JAR `mysql-connector-java-X.X.X-bin.jar` que copiaremos para luego pegarlo en la carpeta *lib* dentro del directorio de instalación de Apache Tomcat.

## Despliegue de la aplicación en Apache Tomcat

Para desplegar la aplicación en Apache Tomcat 6.0.29 bastará con copiar el archivo WAR generado y pegarlo en la carpeta *webapps* del directorio de instalación de Apache Tomcat 6.0.29.

Una vez arranquemos Apache Tomcat 6.0.29, tendremos la aplicación accesible introduciendo en el navegador el nombre del equipo que aloja la aplicación (por defecto *localhost*) + el puerto de escucha de Apache Tomcat 6.0.29 (por defecto *8080*) + *tfc*, es decir: <http://localhost:8080/tfc/>

## Errores frecuentes

La mayoría de errores que se pueden producir se deben a problemas de configuración. A continuación nombramos algunos de los más importantes:

```
GRAVE: Error durante el despliegue del archivo tfc.war de la aplicación web
java.lang.UnsupportedClassVersionError: Bad version number in .class file ...
```

Este error se produce al compilar el proyecto (generar el archivo WAR) con Maven2 usando la JVM versión 1.6 y después desplegar el archivo en Apache Tomcat usando una versión inferior (por ejemplo una JVM versión 1.5). Para solucionar el error, debemos asegurarnos de que Apache Tomcat está usando la misma JVM con la que generamos el archivo WAR, es decir, debe usar la JVM 1.6 de Oracle Sun en su versión 1.6.0\_32.

```
log4j:ERROR setFile(null,true) call failed.
java.io.FileNotFoundException: /path/to/file/tfc.log (No such file or directory)
```

Este error se produce cuando no encuentra el fichero de log y no tiene permisos para crear el árbol de directorios y el fichero. Este error se soluciona creando el fichero en una ruta de directorios de libre escritura y lectura e indicar dicha ruta en los filtros de configuración de la aplicación. Hay que señalar que este error no impide el funcionamiento de la aplicación, pero sí es recomendable solucionarlo para poder hacer un mejor análisis de los logs de la aplicación.

```
ERROR: JDBC Driver class not found: com.mysql.jdbc.Driver
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
```

Este error se produce cuando Hibernate, tratando de crear el pool de conexiones, no encuentra el driver de JDBC para la conexión de Java con MySQL. Para solucionar este error debemos colocar el archivo JAR que hay dentro del ZIP disponible para descargar en la [web de MySQL](#) en la carpeta *lib* de Apache Tomcat.

## Resumen

Para la implantación de la aplicación necesitamos:

- Contenedor de Servlets Apache Tomcat 6.0.29
- Gestor y constructor de proyectos Apache Maven2 en su versión 2.2.1
- JVM 1.6 de Oracle Sun en su versión 1.6.0\_32
- Suite ofimática LibreOffice 3 en su versión 3.5.1.2 o superior
- MySQL JDBC Driver en su versión 5.1.12 o superior

Pasos a seguir:

- Completamos la configuración de los filtros de entorno
- Generamos el archivo WAR de la aplicación con Maven
- Arrancamos LibreOffice en modo servidor
- Desplegamos el archivo WAR en Apache Tomcat

## Conclusiones

---

El proyecto se ha desarrollado según los pasos marcados en la planificación. Las metas que se habían establecido previamente en los objetivos se han cumplido: hemos desarrollado un producto estable que responde a las exigencias del cliente.

Hemos aprendido y asimilado el proceso y las distintas fases por las que pasa un proyecto. Hemos usado un conjunto de componentes externos que hemos tenido que aprender a utilizar y asimilar.

Hemos superado las dificultades que supone la falta de tiempo, con lo que se ha de mostrado el aprovechamiento del tiempo disponible, haciendo más productivas las horas dedicadas al proyecto.

## Glosario de términos

---

Administrador

Usuario registrado avanzado que tendrá acceso a todos los módulos de la aplicación sin restricciones, salvo que aquellas que sean necesarias para mantener la integridad de los datos.

### **Casilla**

Viene referido a la unión de una hora de entra y de salida determinada con un día de la semana determinado.

### **Comunicación bidireccional**

Tipo de comunicación que supone permitir tanto la recepción de respuestas, como el envío de peticiones.

### **Dominio**

Clase Java que mantiene una correspondencia atributo-columna con una tabla de la base de datos. Una tabla *ACTIVIDADES* con una columna *NOMBRE* de tipo *VARCHAR* estará correspondida con una clase Java *Actividades* con un atributo *nombre* de tipo *String*.

### **Framework**

Conjunto de herramientas softwares reutilizables y externas que permiten una mejor organización, estructuración y desarrollo de un proyecto.

### **Manejadores**

Clases Java encargadas de la gestión y proceso de determinadas acciones, en nuestro caso, las peticiones realizadas por el usuario.

### **SGBD**

Sistema gestor de base de datos.

### **Tabla diccionario**

Tabla de base de datos con registros que tienen una baja periodicidad de modificación y de forma puntual. Estos registros no tienen utilidad por sí solos, si no que otras tablas serán dependientes de éstos para completarse. Suele ser un listado cerrado y concreto. Un buen ejemplo de tabla diccionario podría ser la que almacene los tipos de vías posibles: calle, avenida, paseo, carretera, etc.

### **Traspaso de ficheros**

Referido a la migración de datos desde el formato original (papel en nuestro caso) hasta el almacenamiento de los datos en el sistema.

### **Usuario no registrado**

Usuario cuyos datos no están almacenados en el sistema y que, por tanto, no posee unos datos de acceso que le permitan interactuar con la aplicación privada.

### **Usuario registrado**

Usuario cuyos datos están almacenados en el sistema. Posee unos datos de acceso al portal que le permitirán realizar determinadas acciones dentro de la aplicación.