Universidad Complutense de Madrid

Facultad de Informática

Proyecto UCM Reservas

Una herramienta para la mejora de la reserva de espacios de la Facultad de Informática

Trabajo de Fin de Grado en Ingeniería Informática

Curso 2015-2016

Madrid, Septiembre 2016

Autores:

Adriel Saa Romano, Jaime Mayordomo Moreno y Javier Terrón Menoyo

Director:

Iván Martínez Ortiz

Autorización para la difusión y utilización del Trabajo de Fin de Grado y su depósito en el repositorio institucional e-prints Complutense

Los abajo firmantes, alumnos y tutor del Trabajo Fin de Grado (TFG) en el Grado en Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TFG) cuyos datos se detallan a continuación, tanto la propia memoria, como el código, los contenidos audiovisuales, incluso si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Título del TFG: Proyecto UCM Reservas, una herramienta para la mejora de la reserva de espacios de la Facultad de Informática

Curso académico: 2015 / 2016

Nombre del Alumnos:

- Adriel Saa Romano, Jaime Mayordomo Moreno y Javier Terrón Menoyo

Tutor del TFG y departamento al que pertenece:

- Iván Martínez Ortiz (Dept. Ingeniería del Software e Inteligencia Artificial)

Firma del alumno Firma del tutor

En Madrid, a 1 de Septiembre de 2016

“El ordenador nació para resolver problemas que antes no existían”
-- Bill Gates

AGRADECIMIENTOS

En primer lugar, dar las gracias a nuestros padres y amigos que nos han acompañado en todo momento, sin ellos habría sido muy difícil por no decir imposible.

Dar las gracias también a todos los profesores que nos han hecho llegar hasta aquí y a la Facultad de Informática por hacer durante estos años nuestro segundo hogar.

Por último, gracias a Iván Martínez Ortiz por ser nuestro cuarto integrante del equipo y guía en este trabajo. Gracias por todas las cosas que nos has enseñado, las incontables reuniones y sobre todo la paciencia que nos has tenido.

Tabla de contenido

Índice de Ilustraciones

Índice de Tablas

Índice de Abreviaturas

AJAX	*Asynchronous JavaScript And XML*
API	*Application Programming Interface*
ERP	*Enterprise Resource Planner*
CSS	*Cascading Style Sheets*
FDI	*Facultad de Informática*
J2EE	*Java Enterprise Edition*
JPA	*Java Persistence API*
MySQL	*My Structured Query Language*
ORM	*Object Relational Mapper*
POM	*Project Object Model*
SGBD	*Sistema Gestor de Bases de Datos*
SQL	*Structured Query Language*
STS	*Spring Tool Suite*
UCM	*Universidad Complutense de Madrid*
US	*User Stories*

Resumen

En la actualidad, la gestión de espacios comunes en la Facultad de Informática se realiza de diferentes maneras y en ella intervienen diferentes actores dependiendo del espacio que se desea utilizar y la finalidad del mismo. Los espacios utilizados para la docencia (aulas y laboratorios) son gestionados por el Vicedecano de Ordenación académica y Gerencia, las salas de Reuniones, Sala de Grados, Salón de Actos y Despachos de Visitantes son directamente gestionados desde Decanato. Además, la información sobre uso de aulas y laboratorios está disponible y actualizada a través de la web de la Facultad, pero la disponibilidad del resto de espacios no siempre está disponible. Asimismo, el procedimiento de reserva de uno de los espacios comunes (ya sea por motivos docentes o investigadores) requiere hablar o enviar un correo electrónico a la persona que gestiona el espacio primero para conocer la disponibilidad real del espacio (u otros espacios alternativos si no fuera posible reservar el espacio deseado) y por otro lado anotar la reserva para publicarla en la web y para dejar aviso a los ordenanzas del centro para que abran el espacio si fuera necesario. En resumen, la gestión de espacios es un proceso manual y tedioso que se podría agilizar utilizando una herramienta que facilite la gestión de los espacios.

A raíz de estas limitaciones nace "UCM Reservas" como un sistema de gestión de reservas de espacios para la UCM, abarcando la reserva de todo tipo de espacios, así como la asignación de espacios para la impartición de clases durante los horarios planificados para cada cuatrimestre.

En esta memoria se recoge el diseño y la funcionalidad de esta aplicación, así como las tecnologías y procedimientos utilizados para su implementación.

Palabras Clave: aplicación web, reservas, gestión de espacios.

Abstract

Nowadays, the management of common spaces at Facultad de Informática is done in several ways; different actors can take part on it depending on the space and what we want to do on it and it is usually done manually. Spaces used for teaching purposes (classrooms and laboratories) are managed by the Vice Dean of Academic affairs, while meeting rooms, Degrees' Room, Assembly Hall and Visitors' Offices are managed directly by the dean's office. In addition, information about classrooms and laboratories availability is available and updated on the faculty website, however the reservations of the rest of spaces is not always available. Moreover, the reservation procedure of common spaces (for teaching or research reasons) implies to send an email or making a phone call to the person in charge of the management of the space to know the actual availability of the space (or the availability of alternative spaces in case it wouldn't be possible to reserve the desired space), also to note down the reservation for publishing it in the web and to tell the administrative assistants about this reservation, in order to be unlocked if needed. To sum up, spaces management is a manual and tedious process which could be improved by using a software application.

As a consequence of these limitations "UCM Reservas" is devised as a reservations management system for common spaces at UCM, including the reservation of all kind of spaces, as well the allocation of spaces for teaching hours during the the academic terms for each semester.

This report describes the design and functionality of this application, as well the technologies and procedures used for its implementation.

Keywords: web application, reservations, spaces management.

Capítulo 1. Introducción

1.1. Antecedentes y Problema a resolver

La gestión de espacios comunes en la Facultad de Informática sigue diferentes procedimientos e intervienen diferentes actores dependiendo del espacio que se desea utilizar y la finalidad del mismo.

Los espacios utilizados para la docencia (aulas y laboratorios) son gestionados por el Vicedecano de ordenación académica habitualmente ya que se encarga de preparar los horarios del curso académico para todas las asignaturas de las titulaciones que se imparten en la Facultad. En la actualidad, una vez diseña los horarios, los vuelca en dos aplicaciones informáticas, una aplicación que permite publicar los horarios en la web de la Facultad y en el ERP de gestión académica de la Universidad. Durante el curso académico, es habitual que un profesor desee reservar un aula o laboratorio con fines docentes, para aliviar la carga de trabajo en este aspecto, la oficina de la Gerencia de la Facultad se encarga de gestionar estas reservas esporádicas de espacios académicos.

Por otro lado, las salas de Reuniones, Sala de Grados, Salón de Actos y Despachos de Visitantes son directamente gestionados desde Decanato. Su gestión se lleva a cabo a través de la Secretaría de Decanato, llevando el control de la disponibilidad de espacios y canalizando las solicitudes de espacios singulares (Sala de Grados, Sala de Juntas y Salón de Actos) que requieren de aprobación por parte del equipo Decanal y la posibilidad de que una reserva sea anulada o modificada de día por parte de Decanato.

En ambos casos, es necesario interactuar con la persona responsable del espacio para averiguar la disponibilidad y, en su caso, realizar la reserva y aplicar diferentes políticas dependiendo del espacio a reservar. Por tanto, el proceso actual es bastante tedioso, además de no permitir una gestión ágil de espacio (en particular para la reserva de espacios para el día siguiente o para el mismo día).

A raíz de estas limitaciones nace “UCM Reservas” como un sistema de gestión de reservas de espacios para la UCM, abarcando la reserva de todo tipo de espacios, así como la asignación de espacios para la impartición de clases durante los horarios planificados para cada cuatrimestre.

1.2. Solución: UCM Reservas

UCM Reservas es una aplicación web que permite la reserva de espacios en cualquier Facultad en la que estés dado de alta, solucionando así un problema que no solo está presente en la Facultad de Informática sino en muchas otras.

UCM Reservas consigue automatizar un proceso que antes se hacía de forma manual, permitiendo aplicar diferentes políticas de gestión de espacio. En este sentido a través de la herramienta se puede consultar en qué estado (*Confirmada*, *Pendiente* o *Denegada*) se encuentran tus reservas.

1.3. Estructura del trabajo

La presente memoria se encuentra estructura del siguiente modo:

- El ***Capítulo 1*** describe los antecedentes y el problema que resuelve UCM Reservas.
- El ***Capítulo 2*** describe los objetivos y un resumen de los casos de uso de UCM Reservas.
- El ***Capítulo 3*** proporciona una visión de alto nivel de los módulos, pantallas y funcionalidades implementadas.
- El ***Capítulo 4*** se centra en la arquitectura, tecnologías y herramientas utilizadas durante el desarrollo del proyecto.
- En el ***Capítulo 5*** se describe la organización y planificación del trabajo a lo largo de todo el curso académico.
- Los ***Capítulo 6***, ***Capítulo 7*** y ***Capítulo 8*** proporcionan un resumen de las conclusiones, contribuciones y líneas de trabajo futuras.

Finalmente se incluye en el Apéndice A un resumen con todas las **Historias de Usuario** que hemos contemplado para el desarrollo de la aplicación.

Capítulo 2. Objetivos y Casos de uso

2.1. Objetivos

Como se ha mencionado en el capítulo anterior, UCM Reservas aborda la necesidad de facilitar la gestión de reservas de espacios comunes de la Facultad de Informática. Más concretamente, se plantean los siguientes objetivos para el desarrollo de este proyecto:

1. **Gestionar reservas simples**. El sistema permitirá realizar reservas puntuales de un espacio concreto.

2. **Gestionar reservas periódicas**. El sistema permitirá solicitar reservas de un espacio con una cierta periodicidad.

3. **Gestión de reservas complejas**. Se podrán aplicar restricciones a los espacios para que las reservas, tanto simples como periódicas requieran de un paso previo de autorización a su confirmación por parte de los gestores asociados a una Facultad.

4. **Crear grupos de reservas**. El sistema permitirá organizar las reservas en grupos de modo que sea posible visualizarlas y gestionarlas.

5. **Visualizar las reservas**. Permite visualizar las reservas por espacio, grupo o por usuario.

6. **Gestión de espacios y edificios**. El sistema permitirá gestionar los edificios que formen parte de una Facultad y para cada edificio los espacios que tengan.

7. **Soporte para múltiples centros**. El sistema permitirá gestionar varias Facultades, de manera independiente.

8. **Gestión de múltiples espacios de manera sencilla**. El sistema permitirá reorganizar de manera simple (preferiblemente mediante *drag&drop*) las reservas en múltiples recursos. De este modo se permitiría que se pudieran reorganizar las reservas entre los espacios que están disponibles para un centro.

2.2. Casos de Uso

Los diferentes roles que hay en la aplicación definen las distintas acciones que un usuario puede realizar. Estos roles se definen a continuación mediante los casos de uso.

2.2.1. Casos de Uso: Usuario

Los usuarios de la aplicación podrán realizar acciones como iniciar/cerrar sesión, ver y editar sus reservas, o ver y editar sus datos personales. Cada usuario está asociado a un centro, dentro del cual podrá ver los diferentes espacios y sus características y hacer reservas simples o periódicas en su centro.

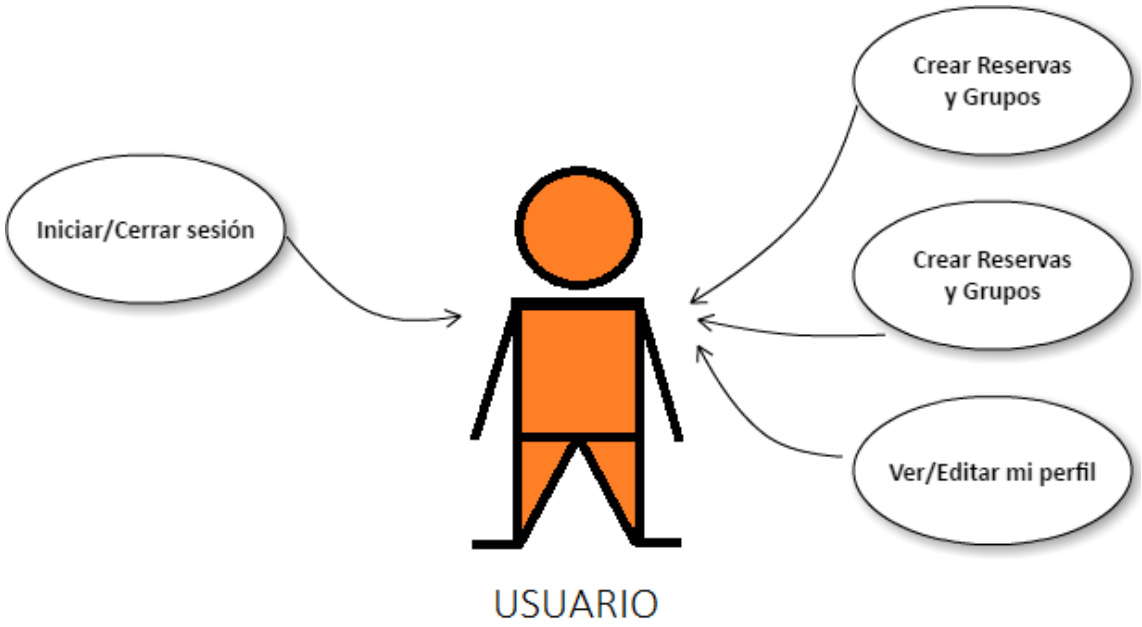

Ilustración 1: Caso de uso de Usuario

2.2.2. Casos de Uso: Gestor

Los gestores estarán asociados a un centro y, además de realizar las mismas funciones que un usuario, podrá cambiar el estado de una reserva y asignar una reserva a otro usuario dentro de su centro.

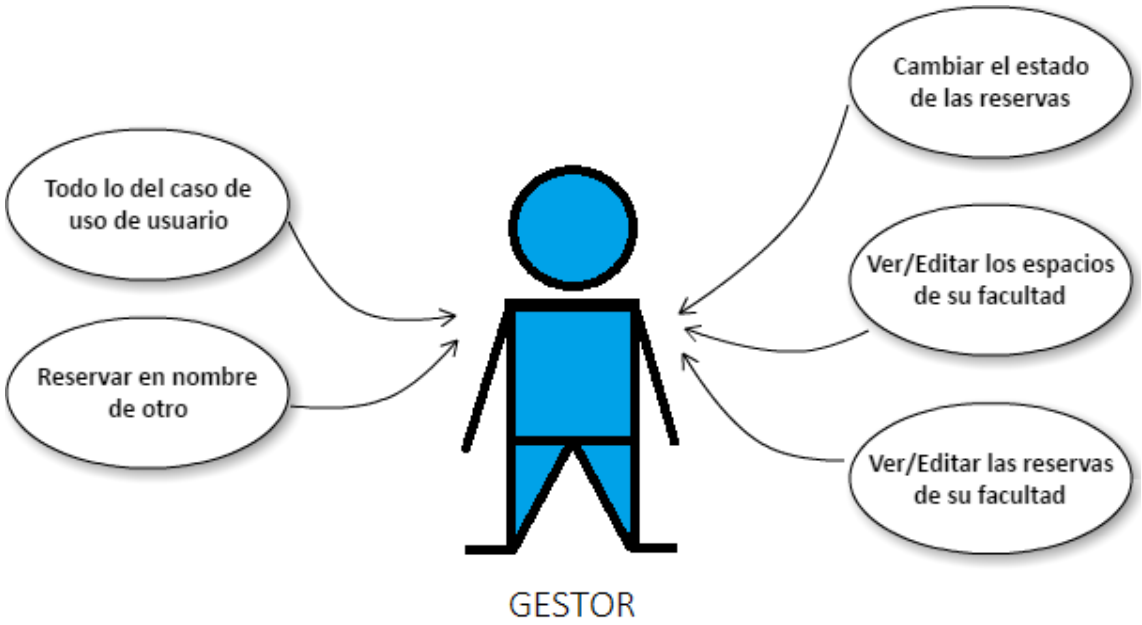

Ilustración 2: Caso de uso de Gestor

2.2.3. Casos de Uso: Administrador

Los administradores, además de realizar las mismas funciones que un gestor, podrán administrar (añadir, editar o eliminar) usuarios, espacios, edificios y facultades.

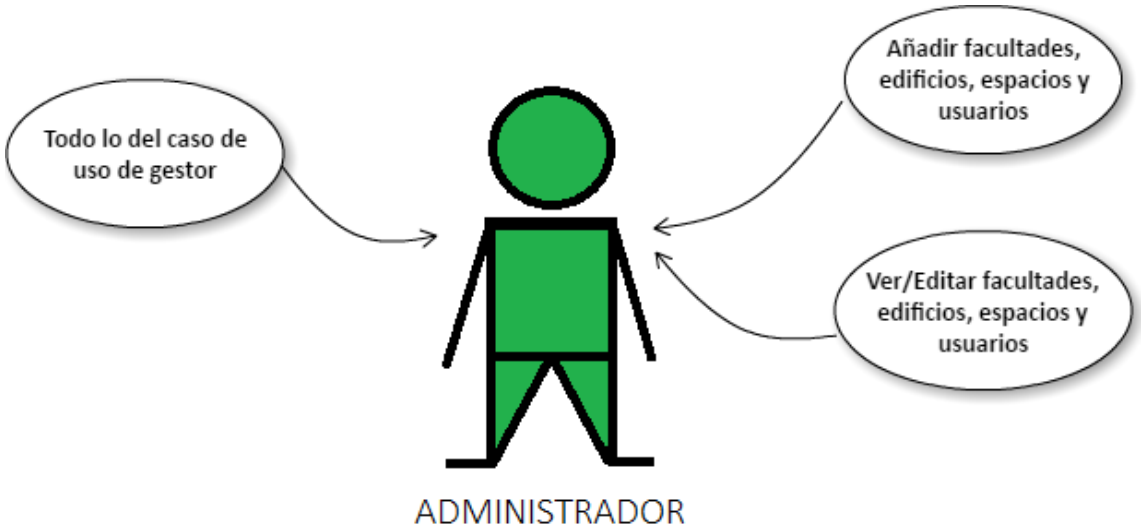

Ilustración 3: Caso de uso de Administrador

Finalmente, ya que durante el desarrollo del proyecto se ha utilizado un método ágil, la elaboración de los casos de uso se ha llevado a cabo mediante el desarrollo de historias de usuario (Referencia). El conjunto completo de historias de usuario implementadas en la aplicación puede encontrarse en el **Apéndice A**.

Capítulo 3. Descripción de la Herramienta

UCM Reservas está compuesto principalmente de dos grandes sistemas, el primero que engloba toda la parte de las reservas y el segundo se encarga de los usuarios. A su vez cada uno está dividido en varios subsistemas.

Ilustración 4: Diagrama general de módulos de la aplicación

En las siguientes secciones se describen con más detalle el modelo de datos que se ha seguido para la implementación de la herramienta, así como la descripción de alto nivel de cada uno de los módulos que la componen.

3.1. Modelo de Datos

Analizando el sistema de reservas de la Facultad de Informática, comprobamos que existían varios métodos para reservar un espacio, y que, dependiendo del espacio, se aplicaban diferentes procedimientos en la reserva. Estas gestiones van desde apuntar el nombre del solicitante en un horario para el día (en una hoja colocada en la puerta del espacio), hasta realizar la reserva a través de la aplicación web que se dispone, pasando por la gestión de la reserva para algunas salas a través de la secretaría de Decanato.

Debido a la variedad de gestiones, decidimos englobar todas las gestiones de manera única, para permitir tener un mecanismo unificado que se encargue de gestionar las reservas. La **Ilustración 5** muestra el modelo de datos de la aplicación.

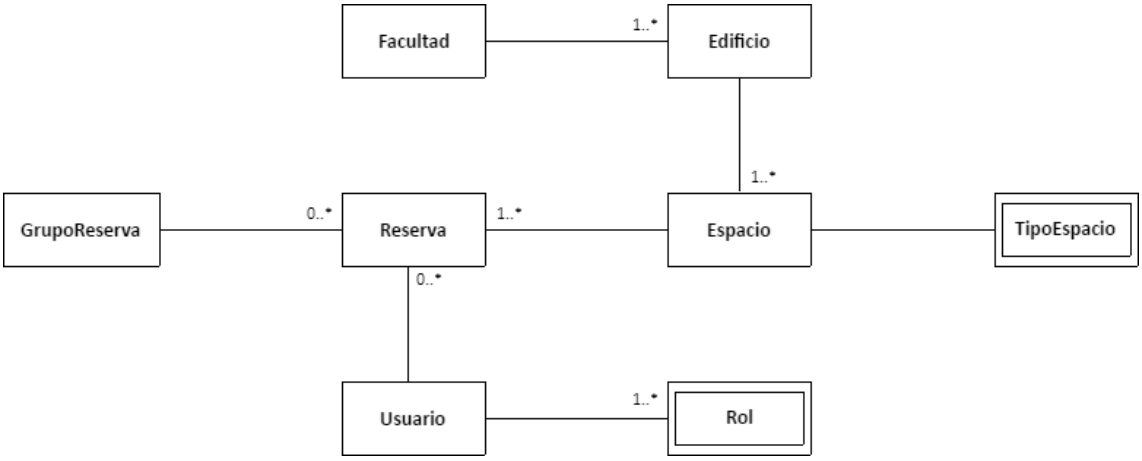

Ilustración 5: Modelo de datos de UCM Reservas

El modelo propuesto se basa en los siguientes ítems públicos:

- **Facultad**: se entiende como una entidad, una facultad puede impartir clases en varios edificios.
- **Edificio**: es el espacio físico donde se sitúan los espacios.
- **Espacio**: Indica cada uno de los compartimentos del edificio donde se puede realizar una reunión.
- **Tipo de Espacio**: especifica el tipo de un espacio, varía en cada facultad. En nuestro caso hay tres, 'Sala', 'Aula' y 'Laboratorio'.
- **Usuario**: representa a cada persona que podrá interactuar con el sistema.
- **Rol**: especifica qué tipos de acciones puede realizar el usuario.
- **Reserva**: señala a cada gestión realizada.
- **Grupo de reserva**: agrupa un grupo de reservas bajo su nombre.

Para mantener el modelo y no perder el histórico de reservas se ha optado por seguir la estrategia de gestión de datos del soft-delete. Esta estrategia consiste en "desactivar" los objetos en vez de eliminarlos. De esta manera se permite recuperar datos borrados anteriormente y no perder información. Es un sistema de seguridad para evitar la pérdida de datos. Esta estrategia se aplicará en los casos de espacios, facultades, edificios y usuarios.

3.2. Módulos de la Aplicación

UCM Reservas está compuesta por 6 módulos que gestionan los diferentes recursos de los que dispone la aplicación. A continuación, se detallan cada uno de los mismos y las funcionalidades que ofrecen a los usuarios.

3.2.1. Módulo de Gestión de Usuarios

Este módulo permite la gestión de usuarios, concretamente la realización de acciones como dar de alta a nuevos usuarios, editarlos o darlos de baja del sistema. También la asignación de roles que dotan a un usuario de privilegios. La **Ilustración 6** muestra la pantalla inicial del módulo, donde se listan los usuarios existentes y a las acciones que son aplicables a cada uno de ellos.

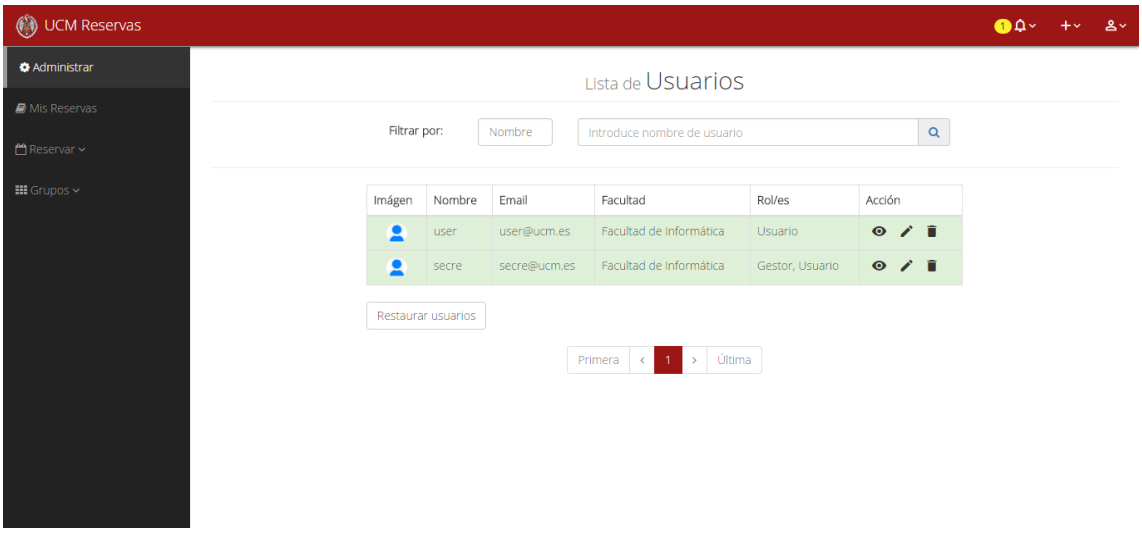

Ilustración 6: Pantalla inicial del módulo de gestión de usuarios

Podemos añadir un usuario rellenando una serie de campos en un formulario (ver **Ilustración 7**).

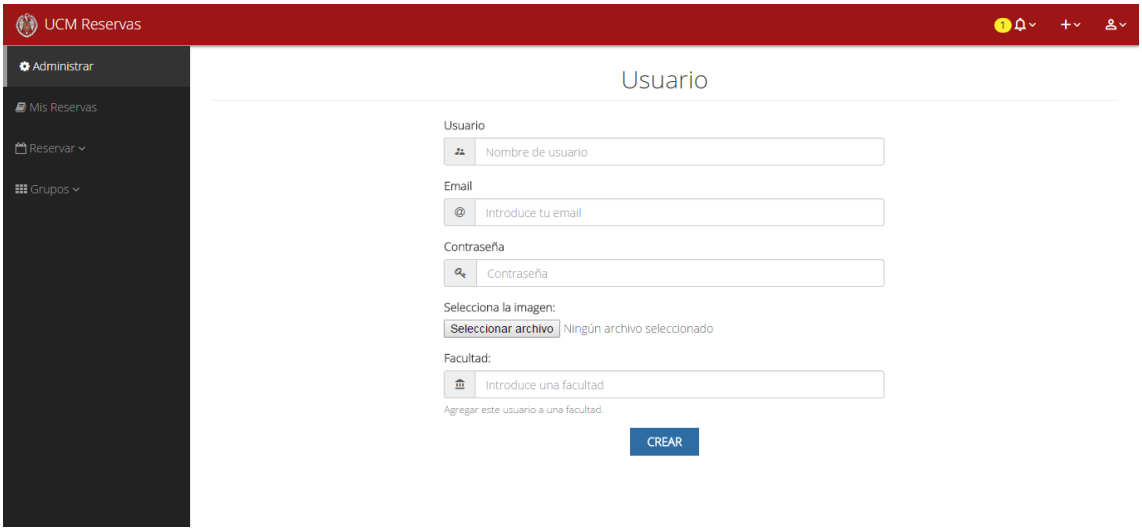

Ilustración 7: Formulario para añadir un nuevo usuario

En caso de necesitarlo, se ofrece la posibilidad de editar los datos del usuario: nombre, correo electrónico, imagen de usuario y asignarlo a una Facultad diferente (ver **Ilustración 8**).

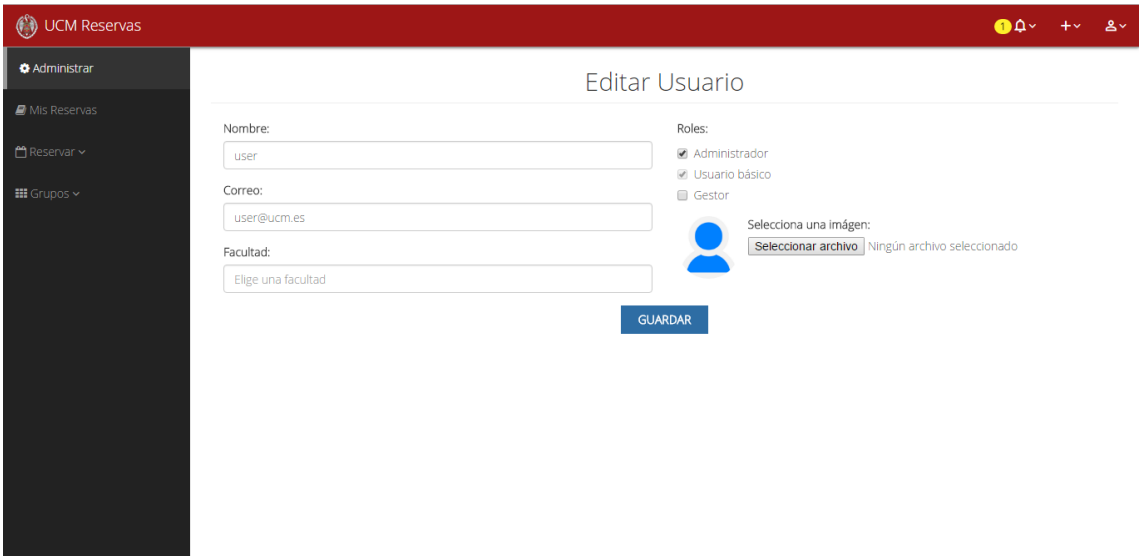

Ilustración 8: Pantalla de edición de un usuario

El administrador también puede restaurar los usuarios que no estén activos.

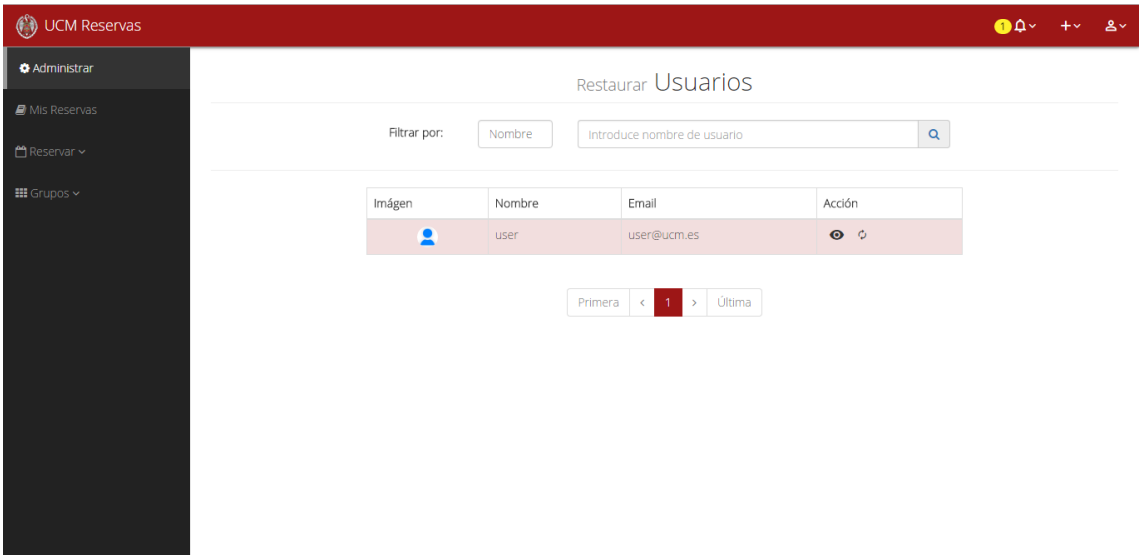

Ilustración 9: Pantalla de restauración de usuarios

3.2.2. Módulo de Gestión de Reservas

Este módulo permite hacer reservas simples o periódicas, editarlas o eliminarlas. Las reservas, además, se pueden añadir a grupos.

A la hora de reservar un espacio lo podemos hacer de dos formas. La primera sería a través de una serie de pasos que consisten en seleccionar el edificio en el que vamos a reservar (por ejemplo, la Facultad de Informática dispone de dos: la propia facultad y el edificio Multiusos). En caso de que la facultad sólo tuviese un edificio, este paso se omitiría.

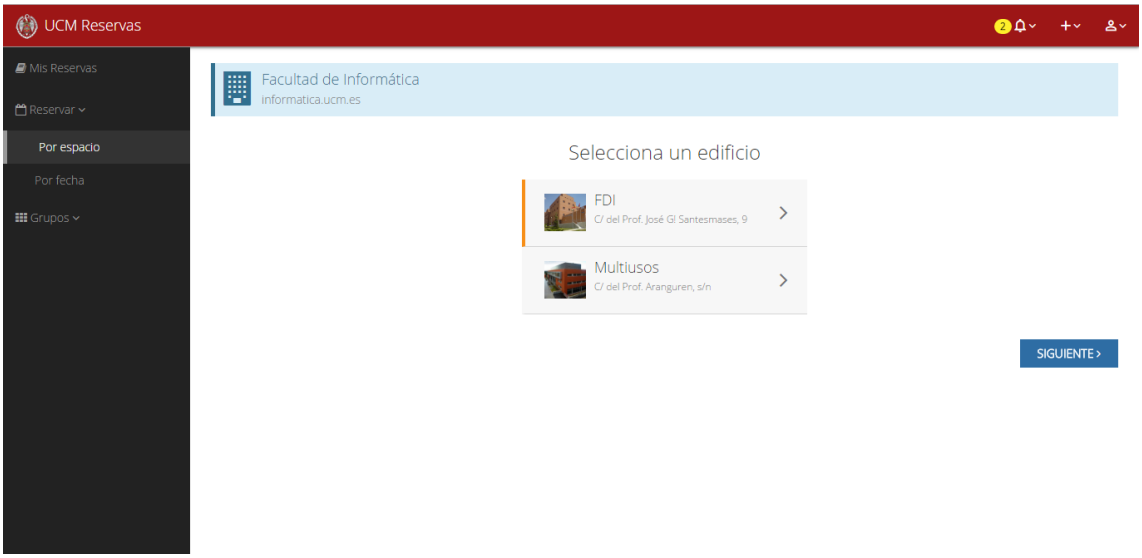

Ilustración 10: Selección de edificio

A continuación, nos aparecerán todos los espacios disponibles en ese edificio (ver **Ilustración 11**).

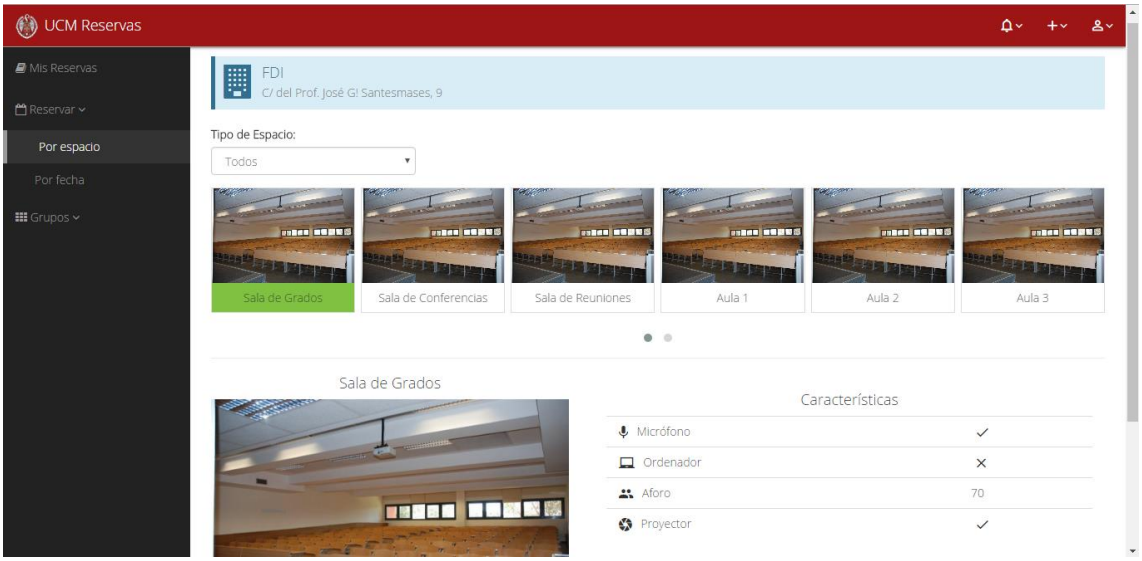

Ilustración 11: Selección de espacio

Al seleccionar uno nos aparecerá un calendario con todas las reservas hechas que tiene (ver **Ilustración 12**). Finalmente seleccionamos una casilla y completamos un formulario que nos aparecerá en un popup para crear la reserva.

La reserva la podemos hacer simple (ver **Ilustración 13**) o periódica (ver **Ilustración 14**), seleccionando la casilla 'Repetir' ampliando el formulario.

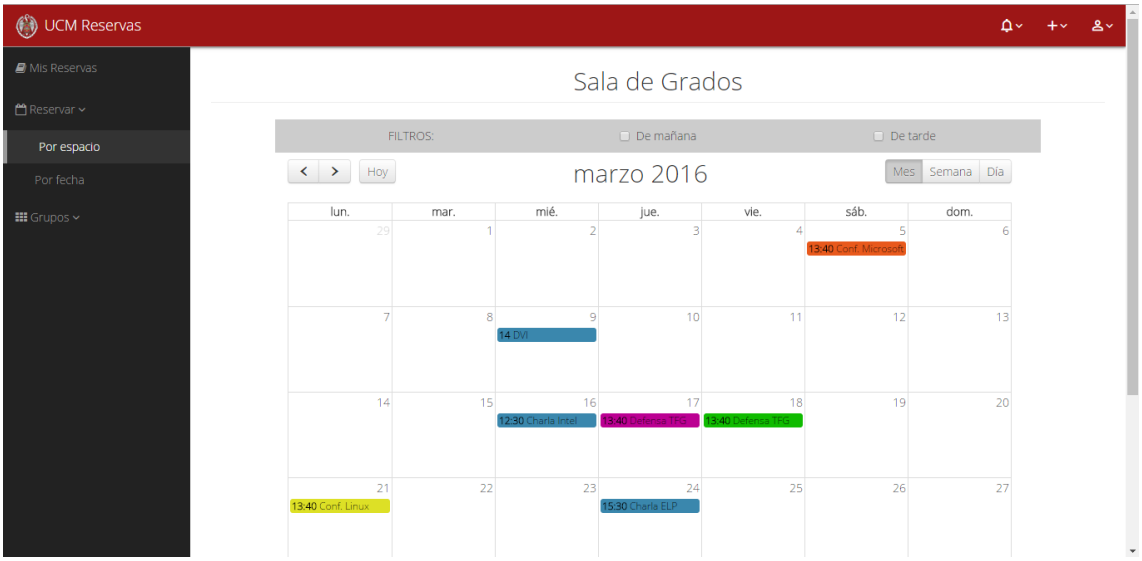

Ilustración 12: Vista de las reservas de un espacio

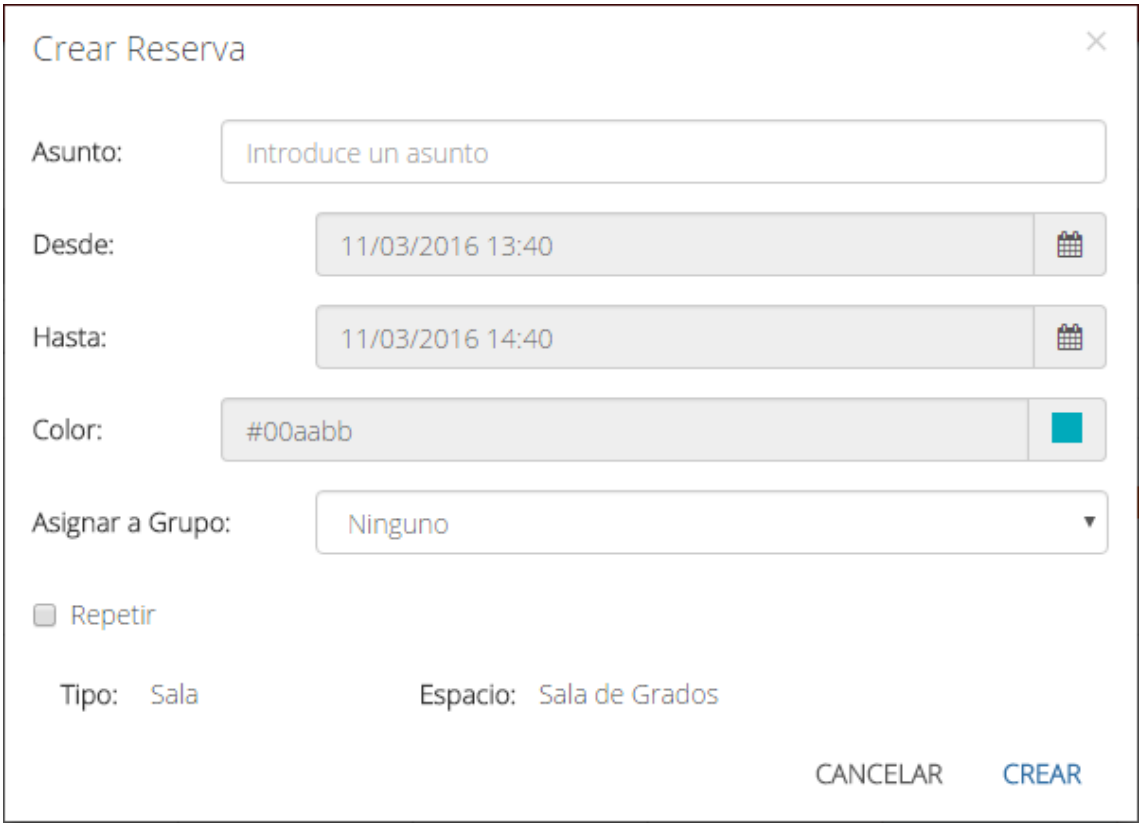

Ilustración 13: Formulario de nueva reserva simple

Ilustración 14: Formulario de nueva reserva periódica

Por otro lado, también es posible realizar una reserva comenzando con la selección de una fecha y un edificio, de modo que la aplicación ofrece qué espacios de ese edificio están libres durante una hora.

Ilustración 15: Búsqueda por fecha

Tanto la edición como la eliminación de reservas se puede hacer desde dos sitios distintos, desde la propia lista (ver **Ilustración 16**) o desde el calendario (ver **Ilustración 17**).

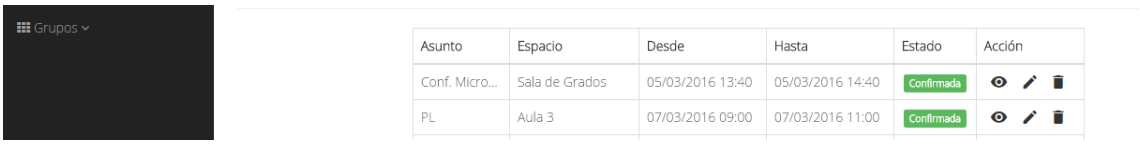

Ilustración 16: Edición o eliminación de reserva desde lista

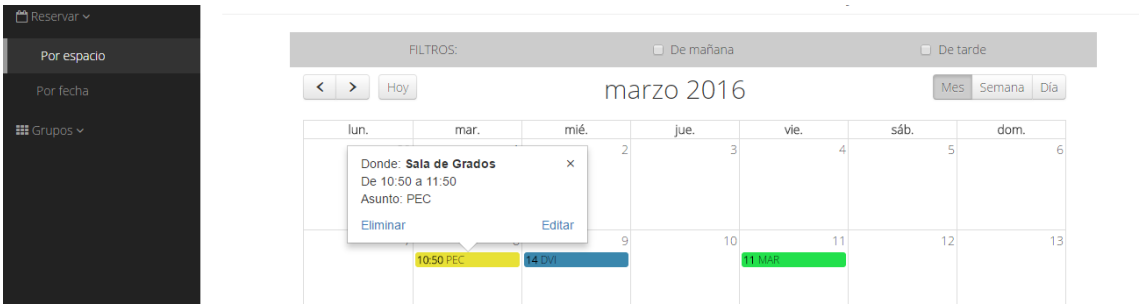

Ilustración 17: Edición o eliminación de reserva desde calendario

Dependiendo de si la reserva a eliminar es simple o periódica nos mostrará un banner (ver **Ilustración 18**) u otro (ver **Ilustración 19**).

Ilustración 18: Eliminar reserva simple

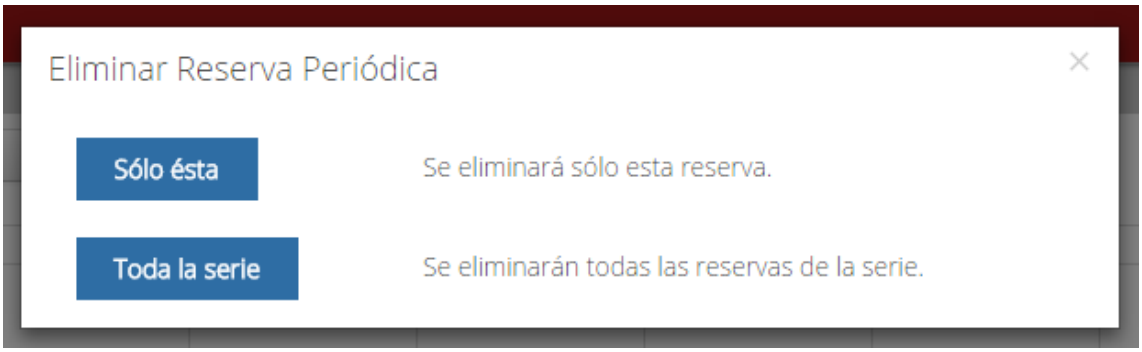

Ilustración 19: Eliminar reserva periódica

El editar una reserva nos permite poder cambiar el asunto, la fecha, el color y el grupo al que pertenece. El gestor, además, tiene la posibilidad de cambiar el estado y el usuario de la reserva (ver **Ilustración 20**).

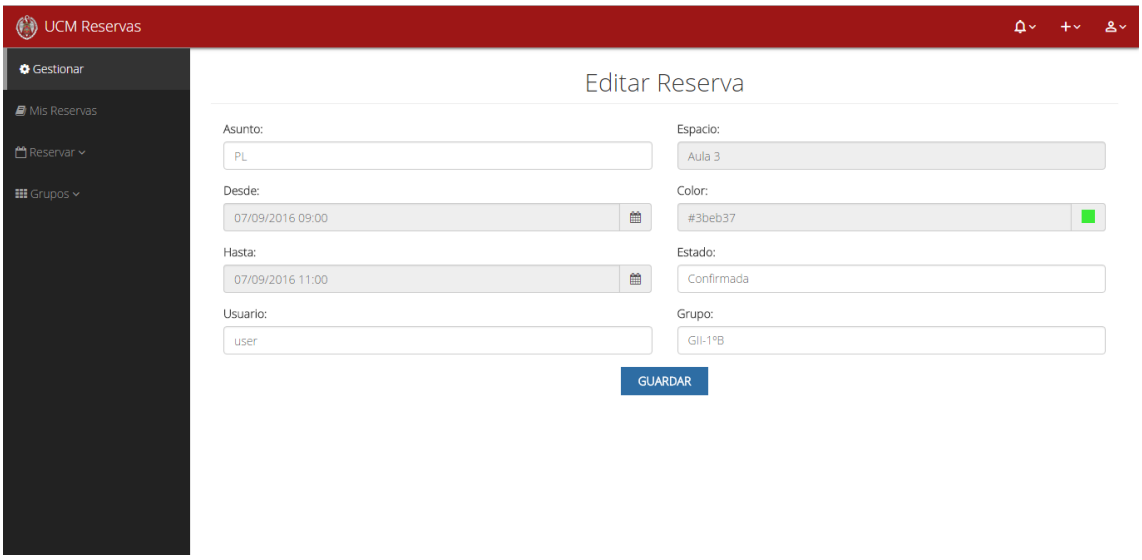

Ilustración 20: Pantalla de edición de reservas del gestor

3.2.3. Módulo de Gestión de Grupo de Reservas

Este módulo permite agrupar las reservas en un calendario aparte. De este modo es posible utilizar este módulo para crear el de cada uno de los grupos de las titulaciones impartidas en la Facultad.

Por ejemplo, se puede tener un grupo ’GII-1ºA’ (Grado en Ing. Informática), y otro ‘GII-1ºB’, con las reservas que tendrán durante todo el año académico.

Nuestros grupos aparecerán situados en el panel de la izquierda siendo accesibles en todo momento. Además, hemos introducido una pequeña búsqueda para evitar el tener que hacer scroll vertical, en caso de que exista una lista larga de grupos, y poder encontrarlo de manera más rápida.

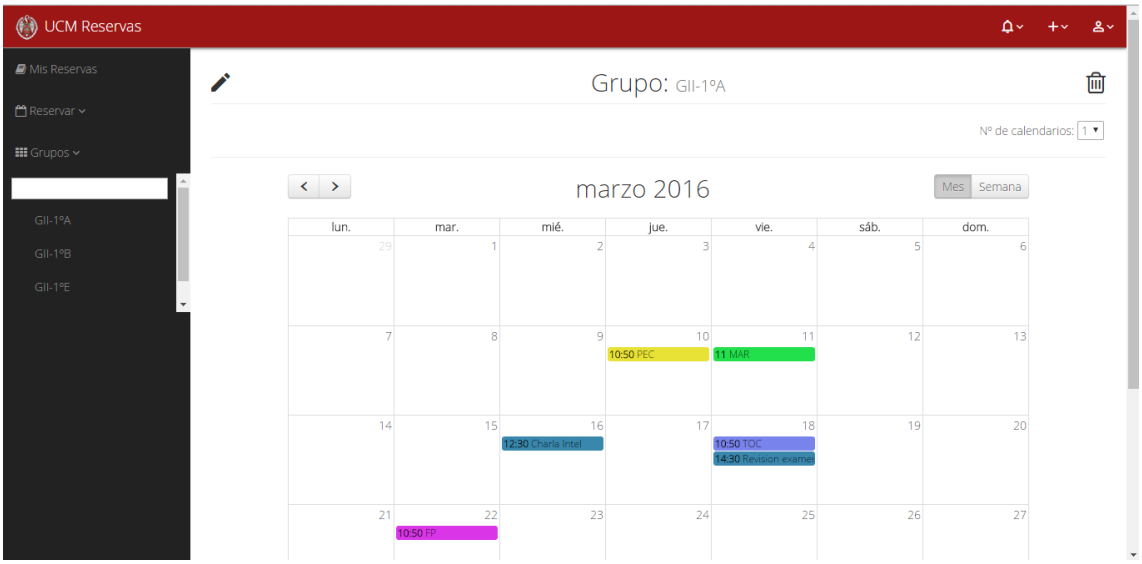

Ilustración 21: Grupo de reservas

Todos los grupos se pueden editar, tanto el nombre del grupo como las reservas que contiene. Puedes eliminar una reserva puntual o cambiar la fecha de una reserva arrastrando y soltando en el propio calendario. Además, es posible moverlas a otro calendario (ver **Ilustración 22**).

También se puede eliminar un grupo, borrando así todas las reservas que hay en él.

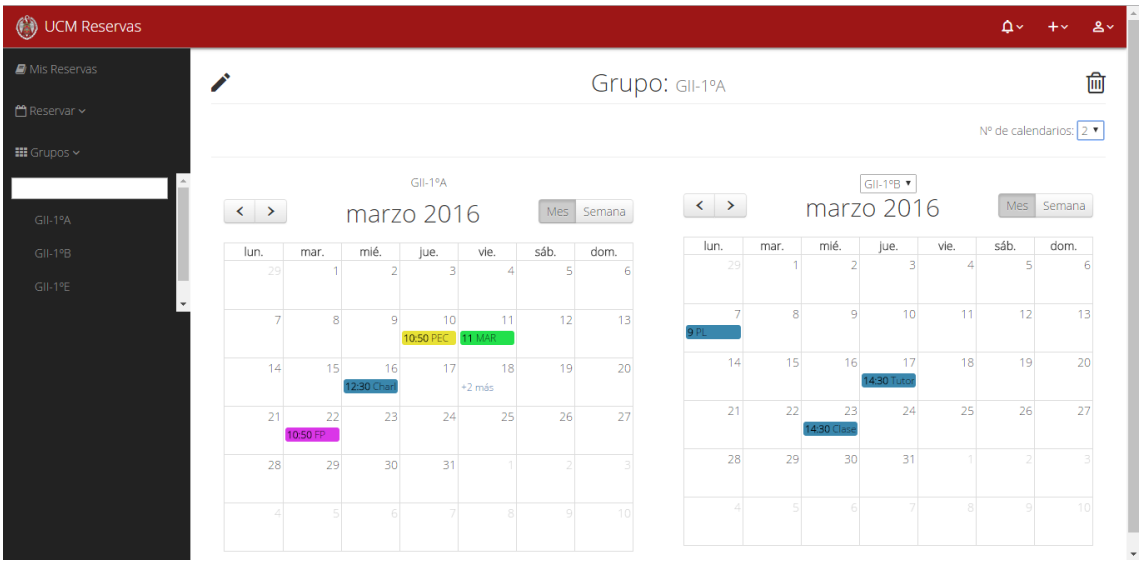

Ilustración 22: Mover reservas entre calendarios

3.2.4. Módulo de Gestión de Espacios

Este módulo permite dar de alta, editar o eliminar espacios pertenecientes a un edificio, así como establecer las restricciones que indican si una reserva solicitada será aceptada automáticamente o precisará de autorización. También se podrá restaurar los espacios previamente borrados. La entrada inicial a este módulo es un listado con los espacios disponibles para la Facultad (ver **Ilustración 23**).

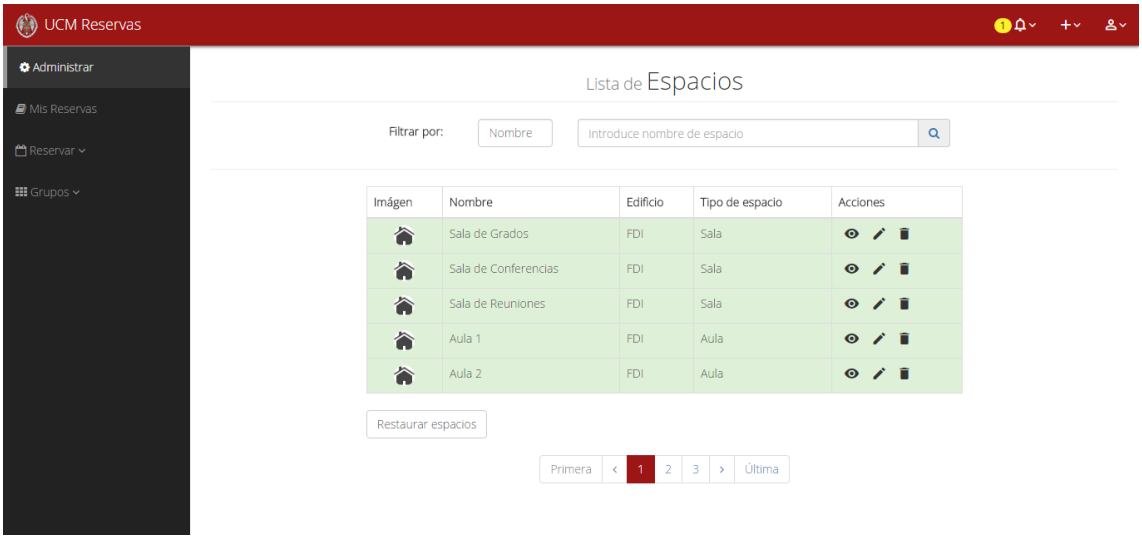

Ilustración 23: Lista de espacios

Los espacios vienen definidos por una serie de características tales como el aforo, si cuentan con un ordenador, y la disponibilidad de proyector y micrófono y que son configurables durante su proceso de creación (ver **Ilustración 24**).

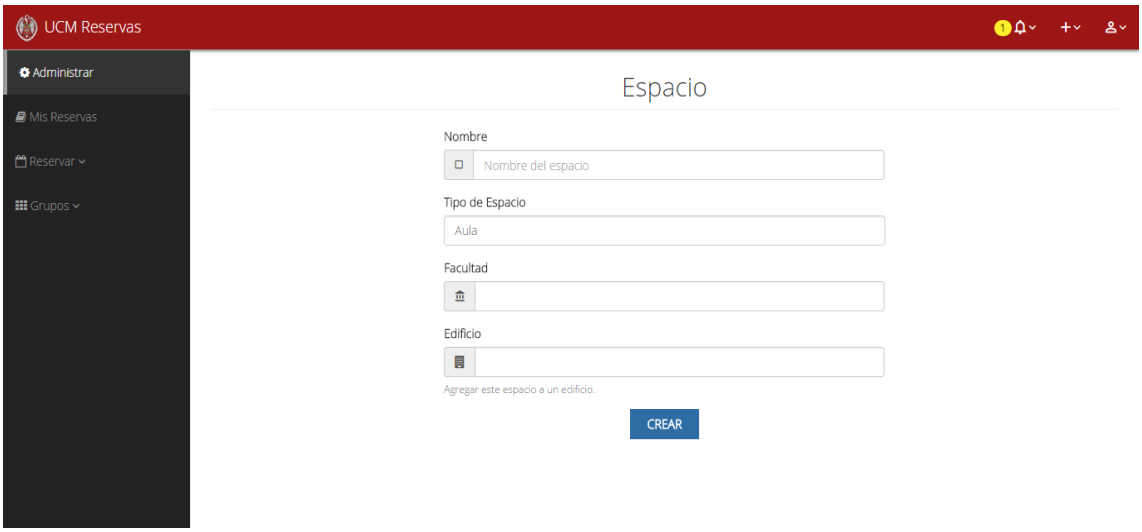

Ilustración 24: Formulario de creación de un nuevo espacio

Durante la edición del espacio es posible establecer restricciones en la solicitud de reservas. En la actualidad, UCM Reservas permite especificar las siguientes restricciones de espacio:

- ***Confirmación necesaria***: un espacio con esta característica requiere que cualquier reserva deba ser confirmada por un gestor o administración de la aplicación.
- ***Tiempo de confirmación***: todas las reservas que no superen el límite de tiempo asignado en esta opción serán confirmadas automáticamente. Si, por el contrario, el tiempo de reserva supera el establecido, la reserva requerirá de la autorización de un gestor/administrador. El tiempo para que se aplique esta restricción podrá modificarlo el gestor o administrador.

3.2.5. Módulo de Gestión de Edificios

Este módulo da la posibilidad de añadir, editar o eliminar edificios pertenecientes a una Facultad.

La información sobre los distintos edificios aparece en forma de tabla (ver **Ilustración 25**), en la cual se muestra solo los edificios activos. Para ver los edificios que no están activos habría que entrar en “Restaurar edificios”.

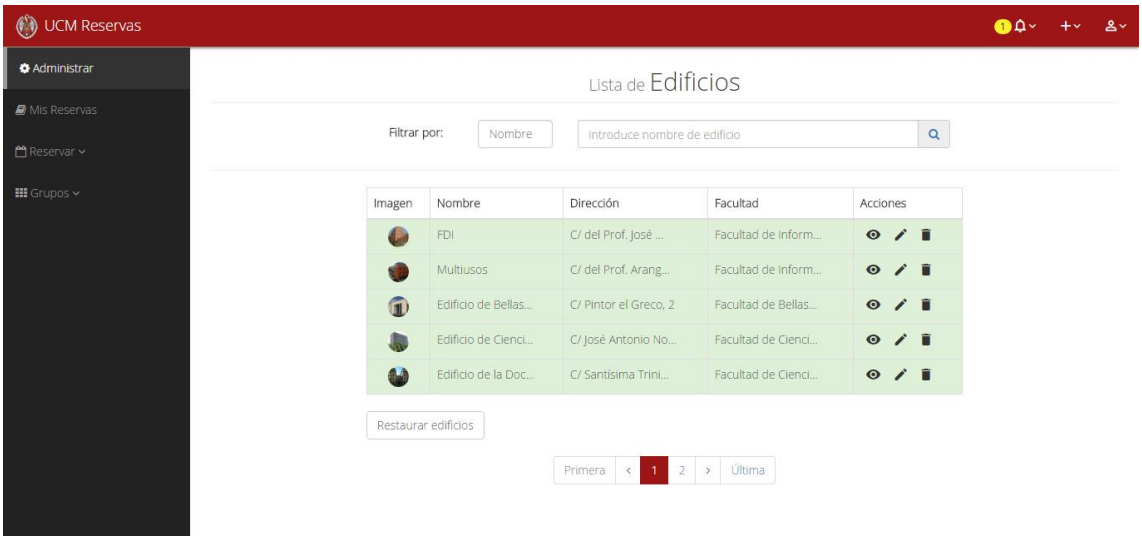

Ilustración 25: Vista principal de la gestión de edificios

Podemos añadir un edificio a un Facultad mediante el formulario que se muestra en la **Ilustración 26**.

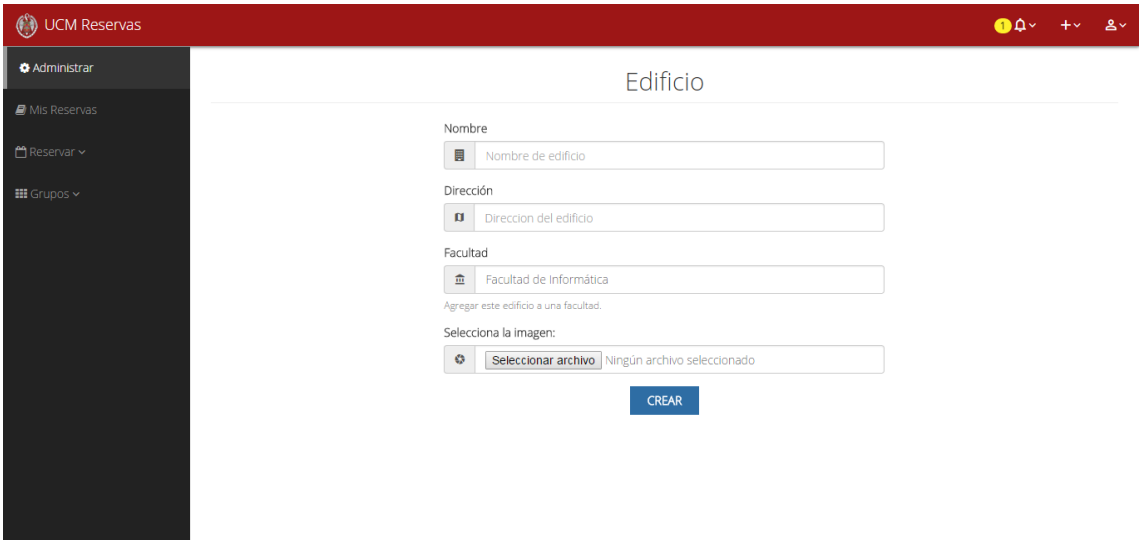

Ilustración 26: Formulario de creación de un nuevo edificio

Al editar el edificio podemos editar cualquiera de los datos del edificio, siempre y cuando éste no se haya eliminado.

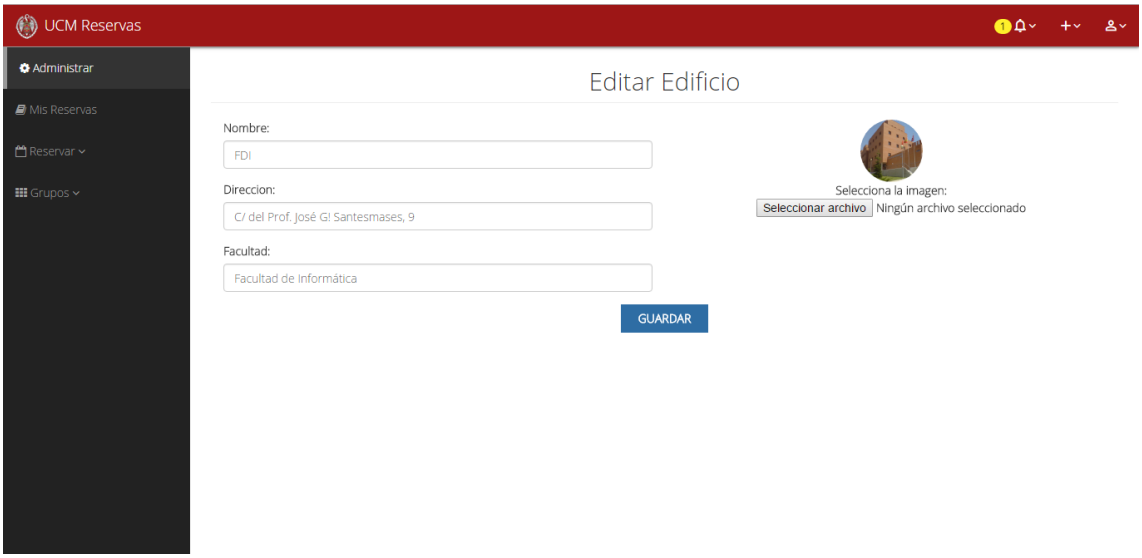

Ilustración 27: Pantalla de edición de un edificio

Además, el administrador también puede restaurar los edificios que no estén activos.

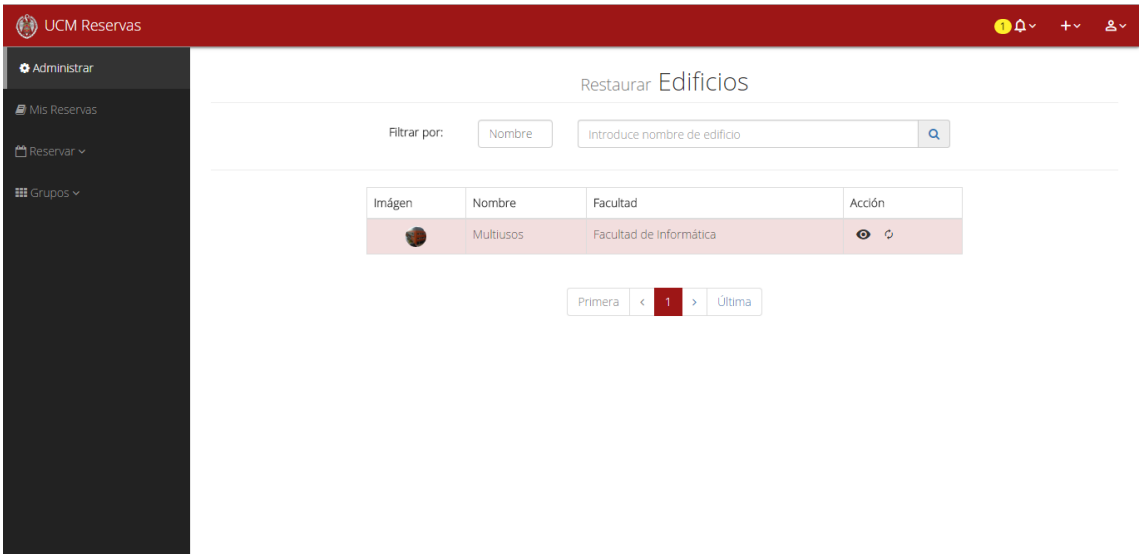

Ilustración 28: Restaurar edificio

3.2.6. Módulo de Gestión de Facultades

Este módulo permite dar de alta, editar o eliminar Facultades dentro de UCM Reservas. De modo similar a otros módulos de la aplicación, la pantalla de entrada (ver **Ilustración 29**) es una lista de todas las Facultades disponibles en la aplicación e incluye las acciones posibles sobre las mismas.

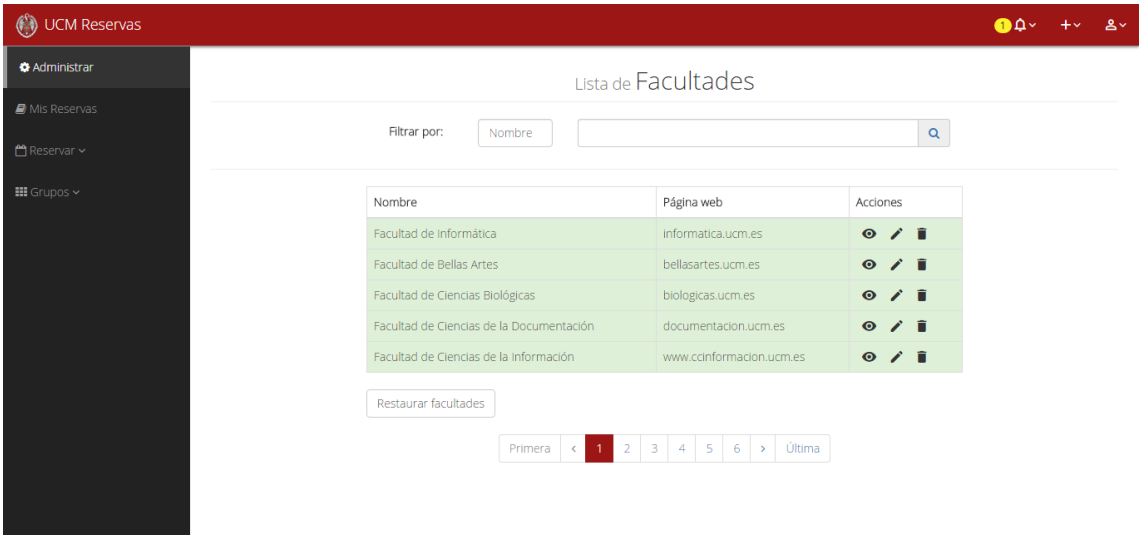

Ilustración 29:Vista principal de la gestión de facultades

La información sobre las distintas facultades aparece en forma de tabla, en la cual se muestra solo las facultades activas. Para ver las facultades que no están activas habría que entrar en “Restaurar facultades”.

Par añadir una nueva Facultad, sólo es necesario definir su nombre y la URL de su sitio web (ver **Ilustración 30**).

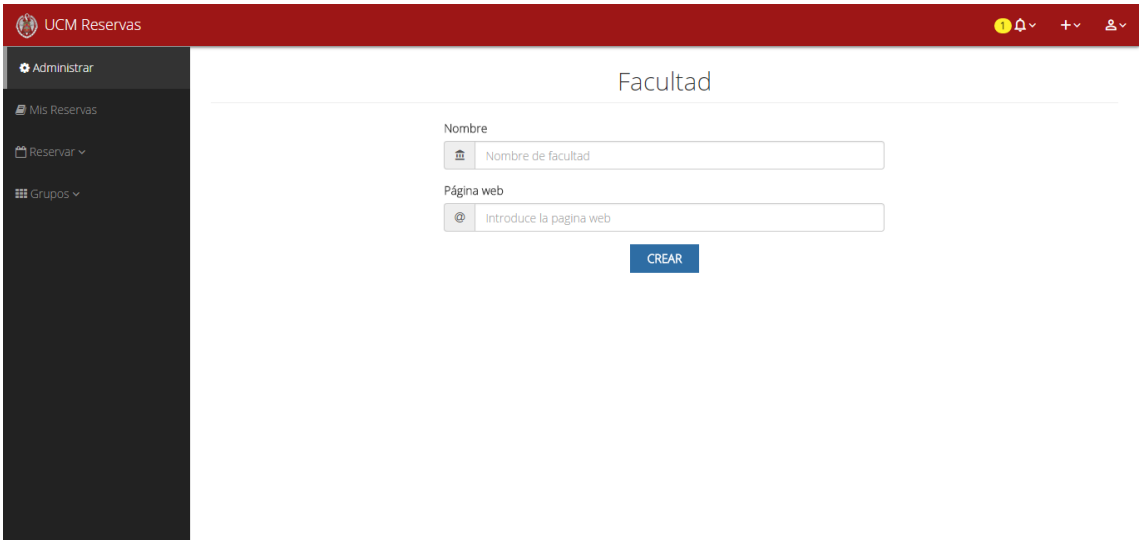

Ilustración 30: Formulario de creación de nueva facultad

Al editar la facultad podemos editar cualquiera de los datos de la facultad, siempre y cuando no se haya eliminado (ver **Ilustración 31**).

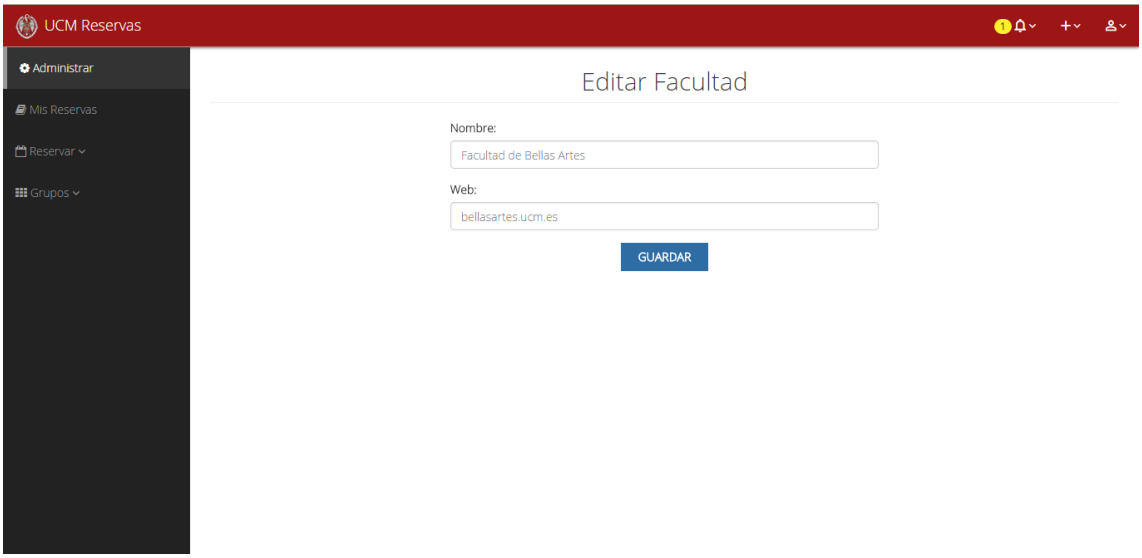

Ilustración 31: Pantalla de edición de una facultad

Finalmente, es posible restaurar las facultades que no estén activas.

Ilustración 32: Restaurar facultad

Capítulo 4. Arquitectura y Tecnología

4.1. Arquitectura de la aplicación

Para el desarrollo de la aplicación se ha optado por la utilización de una arquitectura de varias capas, o multicapa, ya que permite separar la lógica de negocio de la interfaz de usuario y de la capa de datos.

En nuestro caso tenemos tres capas bien diferenciadas que son:

- ***Capa de Presentación (Web)***: en esta capa se encuentran todas las vistas y controladores.

- ***Capa de Servicio (Servicio)***: en ella se encuentran todos los servicios que interactúan con la capa de integración para satisfacer las peticiones del controlador.

- ***Capa de Integración (Repositorio)***: el objetivo de esta capa no es otro sino el de comunicarse con la unidad de persistencia de la aplicación, almacenando o consultando en la base de datos según la petición del servicio.

Cada uno de los módulos descritos en el apartado **3.2 Módulos de la Aplicación** tienen un controlador, el cual permite realizar todas las acciones del usuario. Este controlador dispone de un servicio que se encarga de realizar las operaciones oportunas y de comunicarse con los datos persistidos mediante su repositorio.

Hay casos en los que un servicio necesita información de otros módulos, para ello hace uso de otros servicios que le facilitan el trabajo. El siguiente diagrama muestra con claridad qué servicios se apoyan entre sí.

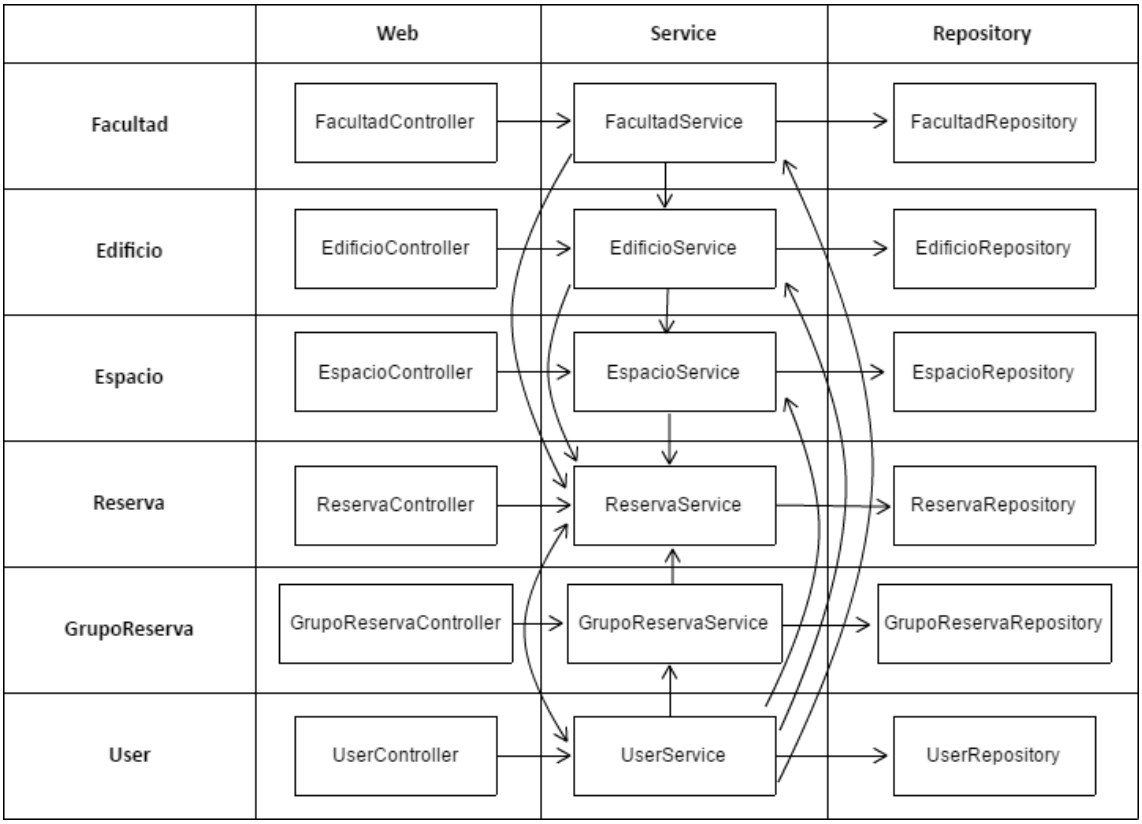

Ilustración 33: Arquitectura de los módulos gestionados

Finalmente, cabe destacar que se ha utilizado el framework Spring para organizar y orquestar toda la aplicación, en particular, por las facilidades que ofrece como la inyección de dependencias entre clases de la aplicación.

4.2. Capa de presentación

La capa web utiliza el patrón MVC (Modelo Vista Controlador). Además de su utilidad y versatilidad este patrón nos ha venido impuesto por el uso del framework Spring que hemos utilizado como base para el desarrollo de la aplicación, en particular Spring Web MVC, que es el módulo utilizado para el desarrollo de la capa web.

El Modelo
Representa la lógica de negocio de la aplicación y es independiente de la vista. Por lo general también se encarga de los datos, pero se ha optado por ponerlos en una capa distinta para lograr una mejor estructuración del código.

La Vista
Se encarga de mostrar los datos al usuario. Éste interactúa con la aplicación produciendo peticiones que le llegan al controlador para posteriormente actualizar la vista.

El Controlador

Hace de puente entre la vista y el modelo. Recibe las peticiones de la vista, las procesa y se comunica con el modelo. Los datos solicitados son devueltos al controlador para poder así actualizar la vista.

4.2.1. API REST

Además de los controladores web clásicos, la aplicación hace uso intensivo de JavaScript en el lado del cliente para agilizar y facilitar la creación y gestión de las reservas. En particular, hace un uso intensivo de un widget de calendario para permitir visualizar y editar reservas existentes. Para poder implementar esta funcionalidad, ha sido necesario la definición de una API REST que permite consultar las reservas existentes o añadir nuevas reservas.

Como parte de la creación de la API REST se ha definido un modelo de datos en formato JSON para representar una Reserva y permitir la comunicación entre el servicio REST y los diferentes widgets utilizados en la aplicación.

4.2.2. Gestión de errores

En toda aplicación es fundamental dar feedback, o respuesta visual, a los usuarios cuando una determinada acción no ha salido como se esperaba. En este sentido en UCM Reservas implementa la gestión de errores mediante las características que ofrece Spring Web MVC que permite tanto proporcionar feedback a los usuarios en la interfaz web clásica o través de la API REST.

Para ello se ha creado la clase **ExceptionHandlerController** que se encarga de capturar excepciones tales como hacer una reserva y que ésta solape con otra, o intentar cambiar tu contraseña de login y que no coincida con la actual. Una vez capturada se crea un objeto **ErrorResult** que contiene un mensaje explicando el porqué de ese error.

Ilustración 34: Error al reservar

4.3. Capa de servicio

Se encarga interactuar con ambas capas, la de presentación y la de persistencia. En esta capa se encuentran los servicios de aplicación y las entidades que implementan la lógica de negocio. Asimismo, las transacciones de la aplicación son gestionadas en esta capa de la aplicación.

Su función es la recibir las peticiones del usuario, es decir, cada acción posible del usuario está representada mediante un método concreto de un servicio de aplicación, que se encarga de coordinar y realizar las operaciones oportunas (delegando responsabilidades a las entidades y a la capa de persistencia).

4.4. Capa de persistencia

En toda aplicación hay datos relevantes que deben ser guardados de alguna manera para su correcto funcionamiento. Para ello hemos optado por utilizar una base de datos.

Existen dos tipos de modelos de bases de datos, las relacionales, que llevan más tiempo en el mercado y por eso son más comunes, y las no relacionales. Ya que la aplicación no tenía unos requisitos sobre rendimiento, nos hemos decantado por el uso de las relacionales.

4.4.1. MySQL

MySQL ha sido el sistema gestor de bases de datos (SGBD) relacionales que se ha usado en UCM Reservas. Los criterios hemos utilizado para elegir han sido:

- Es Open Source.
- Velocidad al realizar operaciones.
- Facilidad de instalación y configuración.
- Usada en otras ocasiones por todos los miembros del grupo.

4.4.2. API de acceso al SGBD y JPA

Para facilitar el acceso a la BD se ha utilizado el ORM que proporciona nativamente la plataforma Java EE y que se implementa a través del framework JPA [9]. JPA simplemente define una API y una metodología para el acceso la BD, pero requiere que se complemente de una implementación de dicha API. De todas las implementaciones existentes que tiene JPA, se ha elegido la de Hibernate ya que además de ser compatible con Spring MVC, tiene la ventaja de ser totalmente independiente del motor de la base de datos que usemos y proporciona características adicionales a la propia API JPA.

4.5. Tecnologías utilizadas en el desarrollo

Como ya se ha mencionado a lo largo de esta memoria, la herramienta UCM Reservas es una aplicación web. En esta sección se introducen brevemente las tecnologías utilizadas en el desarrollo del proyecto.

Java

UCM Reservas utiliza como base principal la plataforma Java. Una de las razones de la utilización de este lenguaje ha sido el amplio conocimiento que teníamos de él, a lo largo de la carrera se han dado varias asignaturas en este lenguaje.

Además de ser un lenguaje orientado a objetos que simplifica mucho el código, existe una comunidad muy fuerte, lo que facilita la búsqueda de documentación.

HTML5

HTML es el lenguaje que se emplea para el desarrollo de páginas web. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. [2]

Esta nueva versión trae consigo varias ventajas respecto a sus predecesores, como por ejemplo que el cierre de algunas etiquetas (img, hr, br o input) ya no sea necesario. Otra ventaja es la inclusión de contenido multimedia sin hacer uso de plugins de terceros.

CSS3

CSS es un lenguaje utilizado mayoritariamente en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente "una página web". Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. [1]

El tener todo el CSS en un archivo aparte nos ayuda a tener un mayor control y limpieza del código HTML. Además, esto nos permite que sea fácilmente modificable y reutilizable en otros proyectos.

Thymeleaf

Se trata de una librería Java que implementa un motor de plantillas de XML/XHTML/HTML5 (también extensible a otros formatos) que puede ser utilizado tanto en modo web como en otros entornos no web [3]. Se ha optado por esta tecnología ya que en conjunto con Spring MVC y ya que el lenguaje es no intrusivo, permitiendo el diseño de las páginas utilizando exclusivamente HTML5 + CSS3 y su previsualización, para posteriormente mediante atributos y etiquetas extra añadir el comportamiento necesario.

JSON

JSON (JavaScript Object Annotation) es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript, aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente [5]. El formato JSON ha sido utilizado como modelo de intercambio entre la API REST y el código JavaScript utilizado en el lado del cliente.

4.5.1 Marcos de trabajo y bibliotecas

Además de las tecnologías base descritas en el apartado anterior, UCM Reservas está implementada utilizando marcos de trabajo y bibliotecas ampliamente utilizadas y soportadas con objeto de simplificar el desarrollo y permitir que la aplicación pudiera evolucionar y cumpliera con garantías de mantenibilidad. A continuación, se describen brevemente las principales bibliotecas y marcos de trabajo utilizados.

Bootstrap

Bootstrap es un marco de trabajo utilizado para el diseño de sitios y aplicaciones web. Contiene plantillas de diseño para los distintos elementos que pueden componer la web (botones, checkbox, cuadros de texto…). Contiene elementos de diseño en HTML y CSS, además de alguna extensión adicional en JavaScript.

La razón por la cual nos hemos decantado por utilizar este framework fue porque que ya lo habíamos utilizado en otras ocasiones. Además, nos permitió hacer la aplicación responsive, o lo que lo mismo, adaptarla a dispositivos móviles.

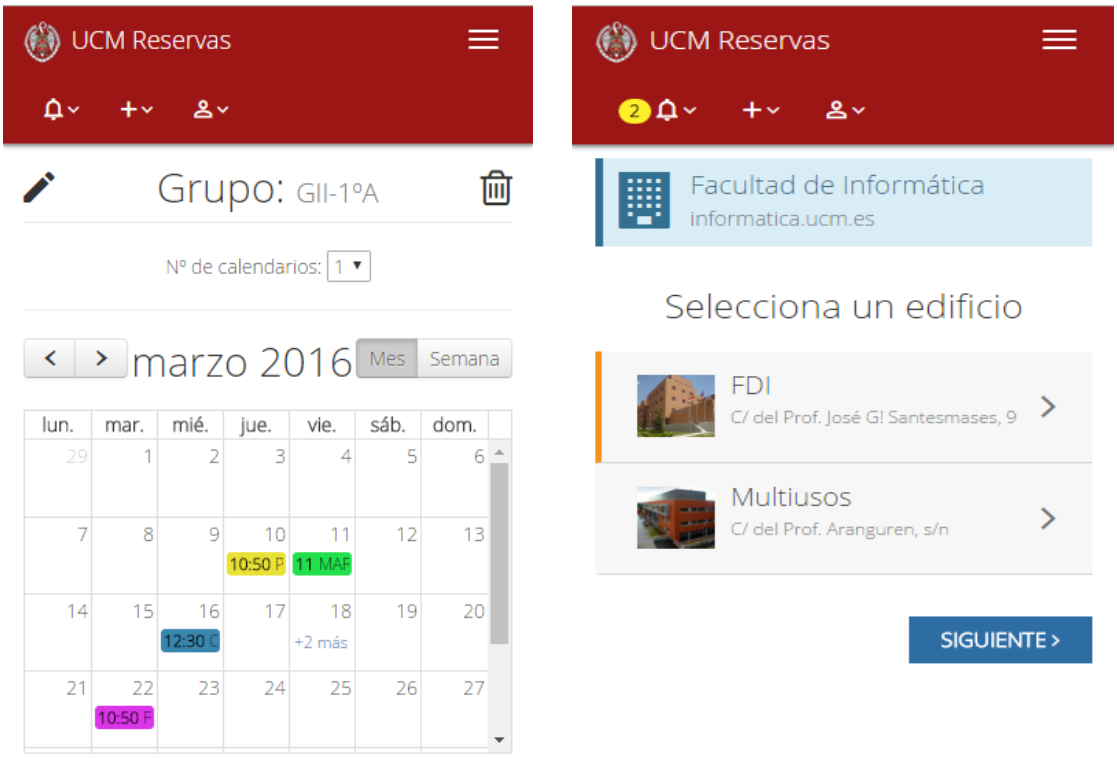

Ilustración 35: Vista móvil grupo reservas y selección de edificio

Font Awesome y Material Design Iconic Font

Por un lado, Font Awesome es un toolkit que incluye iconos y fuentes basado en CSS [6], por otro Material Design Iconic Font proporciona un conjunto completo de iconos y marcas gráficas vectoriales para sitios web. Ambas bibliotecas complementan la funcionalidad ofrecida por Bootstrap.

Nos hemos valido de los recursos que proporciona para hacer reconocibles las distintas acciones que permite realizar la aplicación. nos hemos decidido por este debido a la vistosidad de sus iconos y a su compatibilidad con Bootstrap.

FullCalendar

Se trata de un plugin de jQuery que ofrece un calendario de tamaño completo, con la función de 'arrastrar y soltar'. Utiliza AJAX para traer eventos en marcha cada mes y se puede configurar fácilmente. [8]

Nos hemos decantado por este plugin ya que es muy similar a Google Calendar tanto en apariencia como en funcionalidades.

Moment.js

Es una librería JavaScript que se utiliza, junto a FullCalendar, para el manejo de fechas.

Spring

Spring Framework es una aplicación para el desarrollo de aplicaciones en Java. Se encarga de gestionar las dependencias entre objetos.

Mediante ficheros xml y por anotaciones en las propias clases de Java crea los objetos y los gestiona. Además, mediante ciertas anotaciones permite integrar funcionalidades de otros frameworks, como es el caso de Hibernate, del cual hablaremos a continuación.

Hibernate

Hibernate es una herramienta de mapeo ORM para Java, que facilita el mapeo entre una base de datos relacional y el modelo de datos de una aplicación mediante archivos xml y anotaciones en los beans de las entidades que conforman las relaciones. [7]

Apache Tomcat

Es el contenedor Java EE para Servlets que hemos utilizado ya que al utilizar el marco de trabajo Spring no es necesario utilizar un contenedor Java EE completo.

4.5.1.1 Spring Security y UCM Reservas

La seguridad en las aplicaciones representa uno de los mayores asuntos a tratar en cada proyecto digital. Esta preocupación es mayor si se puede tener acceso a datos sensibles de usuarios. Lo que pretende la seguridad es proteger toda aquella información sensible de la aplicación para evitar el uso incorrecto de esta. En el desarrollo de UCM Reservas se ha establecido una capa de seguridad a través de Spring Security.

La autenticación en una aplicación es un proceso que se encarga de validar los datos de entrada de un usuario. Primero el usuario introducirá sus datos de acceso, normalmente son datos tales como el nombre de usuario o email y su contraseña.

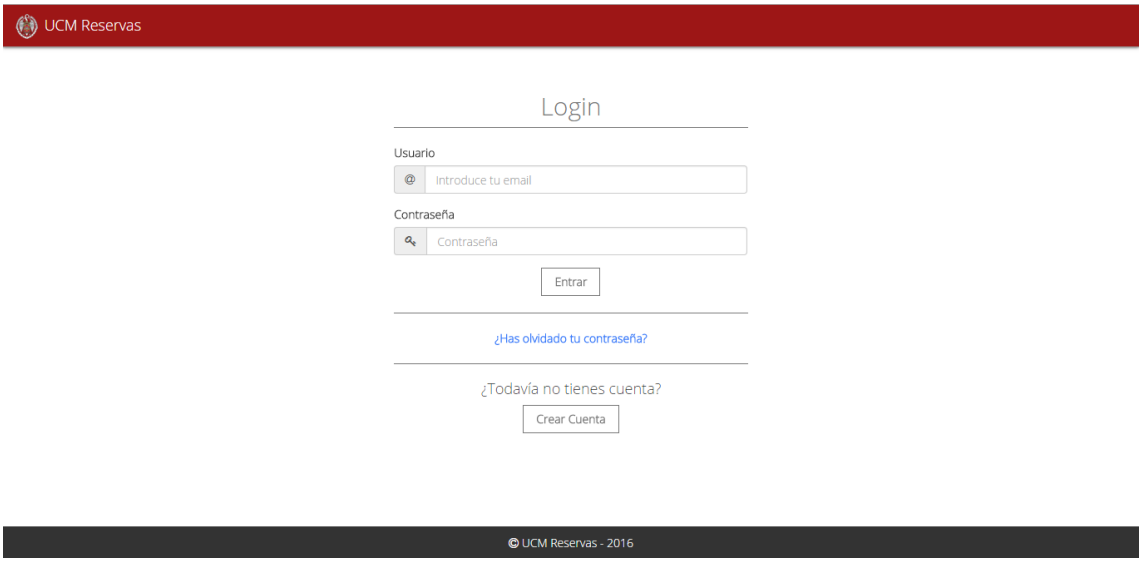

Ilustración 36: Login

La aplicación recibirá estos datos y comprobará si son correctos, es decir, si existen en el sistema. Este proceso de comprobación puede dar dos posibles resultados:

- ***Éxito***: el usuario está dado de alta en el sistema y tendrá acceso la plataforma satisfactoriamente.
- ***Error***: el usuario no ha podido acceder a la plataforma. Ya sea porque haya introducido mal algún dato o simplemente que no esté dado de alta en el sistema.

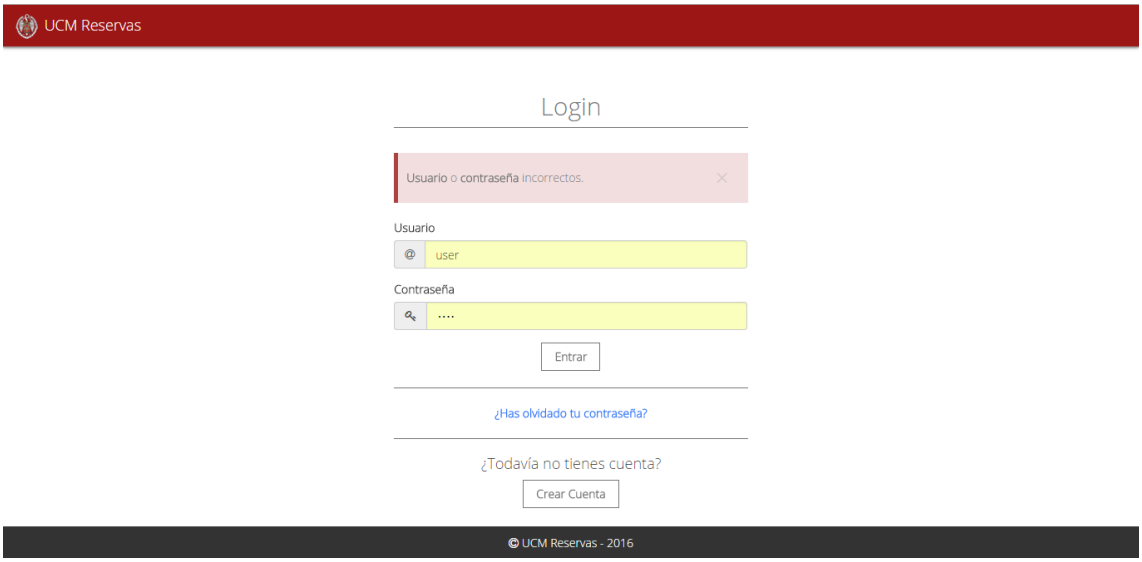

Ilustración 37: Login incorrecto

La autorización en una aplicación es crucial para determinar qué permisos tiene un usuario, es decir, qué acciones se le permite realizar o a qué recursos tiene acceso.

En Spring Security el mecanismo más habitual para gestionar la autenticación y la autorización está basado en el uso de roles. Los roles de los actores identificados para la aplicación (ver ***Capítulo 2***) se han mapeado a roles de Spring Security, más concretamente, definiéndose los siguientes roles:

1. **ROLE_USER**: es el usuario "básico". Puede realizar una reserva, así como editar las reservas que ya haya realizado.

2. **ROLE_GESTOR**: permite gestionar las reservas de una determinada facultad, así como gestionar los distintos edificios y espacios de la facultad que tenga asignada.

3. **ROLE_ADMIN**: es el rol del administrador. Puede realizar reservas, modificarlas, eliminarlas, y se encarga de la gestión global de usuarios, espacios, facultades y edificios.

Estos roles o permisos son acumulativos. Spring Security permite utilizar más de ellos, pero no se han considerado necesarios. En la siguiente tabla se especifica lo que cada rol permite realizar:

	Gestión de Reservas	Gestión de Grupos de Reservas	Gestión de Usuarios	Gestión de Facultades	Gestión de Edificios	Gestión de Espacios
ROLE_USER	SI	SI	SI			
ROLE_GESTOR	SI	SI	SI	SI	SI	SI
ROLE_ADMIN	SI	SI	SI	SI	SI	SI

Tabla 1: Capacidades de los diferentes roles utilizados en la aplicación

4.6. Herramientas de Desarrollo

Las principales herramientas para el desarrollo han sido Spring Tools Suite (STS), y XAMPP.

Ilustración 38: Spring Tool Suite

Spring Tools Suite es un entorno de desarrollo para la plataforma Java y que está especialmente diseñado para el desarrollo de aplicaciones Spring. Este entorno de desarrollo está montado sobre una base de Eclipse y contiene las herramientas de este en relación a Java, Web y Java EE.

Ilustración 39: XAMPP

XAMPP es un servidor independiente de plataforma que contiene herramientas para el desarrollo de proyectos que se usen principalmente con web.

XAMPP es un acrónimo de X (indicando que es independiente de plataforma), A (Apache), M (MySql), P (PHP) y P (Pearl), que muestra algunas de las herramientas que incluye en el paquete.

En el proyecto nos hemos centrado en el servidor Apache Tomcat, y la base de datos MySQL para el control y gestión de la base de datos a través de phpMyAdmin, una herramienta creada en php para poder interactuar con las bases de datos a través de un entorno web.

4.6.1. Herramientas de gestión de configuración

Git, GitHub y SourceTree

Mediante el control de versiones lo que se busca es mantener un historial de cambios en la aplicación por si hubiera que echarse atrás en algún momento y tener una copia de seguridad funcional a la que regresar.

Para esta tarea nos hemos decantado por utilizar Git debido a su versatilidad. Git permite crear ramas independientes entre sí para poder dedicarlas al desarrollo de una característica específica, para luego poder juntarlas todas en una única rama.

Github es un servicio para el alojamiento de repositorios de software gestionados por el sistema de control de versiones Git **¡Error! No se encuentra el origen de la referencia.**. P ara simplificar el uso de Git y múltiples repositorios, hemos utilizado la herramienta Atlassian SourceTree que proporciona un entorno gráfico y amigable para interactuar con Git.

Ilustración 40: Atlassian SourceTree

Maven

Es una herramienta de software para la gestión y construcción de proyectos Java. Maven utiliza un POM para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. [4]

4.6.2. Herramientas de diseño

Como herramienta de diseño se ha optado por ***Cacoo,*** un servicio online para crear, compartir y publicar diagramas.

Ilustración 41: Cacoo [10]

La principal ventaja que ofrece es que, además de ser una herramienta online, no tienes que registrarte para usarla. Tan simple como crear y descargar en el formato que quieras.

Capítulo 5. Diario de Trabajo

En este capítulo se describe brevemente cómo se ha organizado y planificado el proyecto a lo largo del curso.

5.1. Metodología de Trabajo

Se ha optado por una metodología ágil similar a los conceptos de SCRUM, sin seguir esta metodología estrictamente. Se ha ido expandiendo la funcionalidad según se avanzaba en el desarrollo del proyecto.

Puesto que las metodologías ágiles priorizan la comunicación personal, se han mantenido reuniones en las que se han ido obteniendo requisitos adicionales y se han comentado los avances de la aplicación. Estas reuniones se han establecido de manera semanal, teniendo que retrasar el encuentro a 2 semanas en casos aislados.

En cada reunión se comentaba los avances obtenidos en el estado del proyecto, los problemas que se han encontrado y como se podrían solucionar. Así mismo se fijaban unos objetivos para las siguientes reuniones, incluyendo la solución de los problemas y la implementación de nuevas funcionalidades.

5.2. Organización y planificación

En las metodologías ágiles que hemos seguido hay unas fases que se desarrollan para cada iteración realizada. A continuación, explicaremos en qué consiste cada una de estas fases:

1. **Fase de Planificación**
   Esta fase consiste en la identificación y priorización de la nueva funcionalidad a desarrollar y el análisis del impacto que tiene estas nuevas funcionalidades en la versión actual de la aplicación.

2. **Fase de Análisis de requisitos**
   Una vez llegados a este punto se detallan los requisitos de alto nivel a solucionar en la iteración actual. Pueden surgir requisitos que se apliquen a iteraciones futuras. También se obtendrá un esquema de los módulos necesarios para el proyecto. En particular durante esta fase se desarrollan las historias de usuario que guiarán el desarrollo y la validación de las nuevas funcionalidades.

3. **Fase de Diseño**
   Aquí se obtendrá la interconexión de servicios para obtener la solución a los problemas detallados por los requisitos y las historias de usuario.

4. **Fase de Codificación**
   En este punto se crean las clases y métodos necesarios especificados por la fase de diseño y se implementan. Al ser un grupo, las tareas se reparten entre los miembros, intentando que los objetivos se separen en tareas aisladas de otras. También se comprueba que la estabilidad del producto se mantenga intacta.

5. **Fase de Revisión**
   Cuando se ha terminado la codificación se realiza esta fase para comprobar que la implementación es adecuada y que no se han introducido errores en las características que ya están implementadas.

6. **Fase de Documentación**
   Finalmente se anotan todos los avances del proyecto en un documento. Esto viene muy bien para recopilar la información y poder revisar donde se encuentra un conflicto en caso de que al extender la funcionalidad se modifique alguna característica que se encuentre implementada en algún paso anterior. Así mismo, permite empezar la documentación final entregable con gran parte anotada y desarrollada.

Los primeros meses se emplearon como toma de contacto de las tecnologías y herramientas a emplear en el desarrollo del proyecto, haciendo hincapié en el software para el control de versiones Git a través del entorno visual de SourceTree, y de la configuración y uso de las características de Spring en su entorno de desarrollo Spring Tool Suite.

5.3. Diario mensual de desarrollo

A continuación, se detallan las actividades que se han realizado en las reuniones que hemos tenido a lo largo del curso.

Fecha: Septiembre - Octubre

- Análisis y comprensión de las tecnologías y herramientas a utilizar a través de documentación y tutoriales.

Tabla 2: Diario de desarrollo Septiembre - Octubre

Fecha: Octubre - Noviembre

- Primera toma de contacto con Spring Framework y Github.
- Desarrollo de proyecto enfocado a familiarizarse con las herramientas de desarrollo.

Tabla 3: Diario de desarrollo Octubre - Noviembre

Fecha: Noviembre - Enero

- Creación de prototipo plano en HTML.
- Primer prototipo de interfaz funcional.

Tabla 4: Diario de desarrollo Noviembre - Enero

Fecha: Enero - Marzo

- Implementación de los módulos troncales. Usuarios y Reservas.

Tabla 5: Diario de desarrollo Enero - Marzo

Fecha: Marzo - Mayo

- Implementadas nuevas funcionalidades: añadir, editar y eliminar reservas periódicas.
- Se añaden las vistas del gestor y del administrador.
- Implementación de funcionalidades básicas para gestor y administrador.

Tabla 6: Diario de desarrollo Marzo - Mayo

Fecha: Mayo - Junio

- Recopilación de la información de las fases de documentación realizadas.
- Extensión de las funcionalidades de gestor y administrador.

Tabla 7: Diario de desarrollo Mayo – Junio

Fecha: Junio - Agosto

- Modificaciones en la memoria.
- Extensión de funcionalidades de la aplicación

Tabla 8: Diario de desarrollo Junio - Agosto

Fecha: Agosto - Septiembre

- Primera versión de la memoria.
- Revisión final del código de la aplicación.

Tabla 9: Diario de desarrollo Agosto - Septiembre

Fecha: Septiembre

- Versión definitiva de la memoria.
- Preparación de la presentación del proyecto.

Tabla 10: Diario de desarrollo Septiembre

Capítulo 6. Conclusiones

La gestión de aulas en la Facultad de Informática de la Universidad Complutense de Madrid siempre ha supuesto un esfuerzo superior a la hora de realizar reservas y organizarlas. Esto ha supuesto una necesidad de mejora en la aplicación dedicada a este objetivo.

Debido a la modernización de las herramientas de diseño, especialmente en los últimos años, donde se pretende que toda la visualización sea lo más simple posible, se ha puesto el esfuerzo para simplificar la gestión y hacer más amigable la interfaz visual.

6.1. Resumen de Contribuciones

El primer objetivo de la aplicación era conocer la situación y modo de gestión de las reservas dentro de la Facultad de Informática. Esto nos hizo realizar un análisis a fondo de los mecanismos y herramientas destinadas a gestionar los diferentes espacios, así como buscar posibles medidas de mejora de estas.

Debido a las necesidades de profesores y la disponibilidad de los administrativos se ha buscado realizar una aplicación que permita crear reservas automáticamente en la mayoría de ocasiones siguiendo un conjunto simple de reglas que además permitiera la intervención del personal administrativo para las reservas estas reglas.

Otra funcionalidad a destacar es la posibilidad de modificar las reservas directamente desde un calendario propio de la aplicación, en el cual se muestran las reservas que tendrá en las horas y días cercanos.

Se ha pretendido que sea posible la extensión de la funcionalidad de la aplicación a otras facultades, para poder mantener un estándar dentro de la misma institución. Cada facultad podrá poner sus propios criterios a la hora de realizar una reserva.

Hemos de destacar las técnicas aplicadas al desarrollo de sistemas interactivos, que nos han permitido crear una interfaz visual amigable y agradable de uso sencillo, y las técnicas de desarrollo ágil de ingeniería del software, que han permitido afrontar de manera rápida y ordenada el desarrollo del proyecto.

Capítulo 7. Conclusions

Space management at Facultad de Informática of Universidad Complutense of Madrid have been a difficult task, usually this task required great effort when doing and/or organising different reservations. Hence, a software application that helps with this process will be helpful.

The development of design tools during recent years have been focused on simple visualization, hence a great effort has been made in order to simplify the management of reservations and making a modern and user-friendly UI.

7.1. Summary of contributions

Our first objective was to understand the reservation process used at Facultad de Informática. This led us to do an analysis of the different processes and the used software applications to support them.

In order to fulfill professors' needs and the availability of administrative staff, we have created a software application that follows a simple set of rules allowing the professors to make automatic reservation in most cases, but allowing the manual intervention of administrative staff when required.

Another remarkable functionality is the ability of modifying the reservation through a visual calendar in a simple way, allowing the user to visualize the already committed reservations for the same space in the following days.

The application has been developed in a way that it is also possible to be used directly by other faculties, although it is possible to choose its own reservations' restrictions by each faculty.

We have to highly remark the interactive systems development and software engineering techniques applied that facilitated us to create a friendly interface, easy to use, and which allowed us to tackle the development of the application in a faster and sorted way.

Capítulo 8. Trabajo futuro

8.1. Integración de SSO con Google

En toda aplicación siempre hay una parte que no se puede evitar, el registro previo a loguearse. El proceso cada vez es más corto, pero sigue sin gustarnos.

Lo ideal para este caso, ya que todo el personal docente de la UCM dispone de una cuenta de google, sería el poder autenticarnos con ella.

8.2. Otras mejoras

A lo largo del desarrollo de la aplicación han ido surgiendo posibles mejoras que no se han podido implementar por falta de tiempo. Veamos cuales:

Invitar a usuarios a una reserva

Se trata de dar la posibilidad al usuario que reserva de poder invitar a más usuarios, de esa manera las personas que quieran asistir tendrán una copia de esa reserva en su calendario como recordatorio, en caso de que acepten dicha invitación.

Establecimiento de cookies para no tener que loguearse cada vez

Se puede añadir un sistema que recuerde al usuario logueado, si este no ha abandonado la aplicación mediante el botón de logout. Así, el usuario que quiera volver acceder sin haber cerrado su sesión podría volver a acceder a esta sin necesidad de volver a introducir usuario y contraseña.

Extensión a distintas universidades

Podría ampliarse la funcionalidad en el sentido de extensión del ámbito del producto, haciendo que este pueda usarse en otras universidades igualmente.

Mapas interactivos

Se podrían incluir planos interactivos de cada planta del edificio, pudiendo seleccionar cualquier espacio y acceder así a sus reservas. Esto nos permite ver todos los espacios de manera más rápida y además, saber exactamente donde está situado.

Eliminación automática de reservas

Esta mejora surge de la idea de mejorar el rendimiento de la base de datos y consiste en eliminar las reservas de la base de datos que estén 'caducadas', es decir, cuya fecha de finalización ya haya pasado. De esta manera al haber menos datos las consultas se harían mucho más rápidas.

Contribuciones individuales

A continuación, se describen las contribuciones que cada miembro del grupo ha aportado al proyecto.

Adriel Saa Romano

Me he encargado de diseñar e implementar la interfaz de usuario. También he llevado a cabo la implementación de las reservas, es decir, creación, edición y eliminación de reservas simples, reservas periódicas y grupos de reservas.

He realizado las distintas formas que tiene la aplicación para reservar: reservar por espacio y reservar por fecha. Además, he implementado una búsqueda rápida para los grupos del usuario.

Otras aportaciones han sido: la autenticación de usuarios, la edición del perfil, control de errores en login, reservas, grupos y perfil, así como el paso de reservas entre varios calendarios con la finalidad de la creación del horario de la facultad.

Jaime Mayordomo Moreno

Me he encargado del diseño y la implementación de la interfaz de administrador. Además, he realizado la implementación de la gestión de usuarios, facultades, edificios y espacios, que engloba la creación, edición y eliminación de usuarios, facultades, edificios y espacios.

He incluido, para cada unidad gestionada, un filtro con el objetivo de acelerar la gestión en el modo administrador. También he implementado la tecnología "soft-delete", utilizada a la hora de activar o desactivar usuarios, facultades, edificios y espacios.

Javier Terrón Menoyo

Me he encargado del diseño y funcionamiento de las funciones del gestor. Me he encargado de la edición, creación, borrado y restauración de usuarios, espacios y edificios. También me he encargado de la edición de reservas y de la implementación de las restricciones en los espacios para indicar si fuese necesaria una autorización para las reservas que se realicen en dichos espacios.

He contribuido en la creación de un módulo de inicio donde indica la dirección a la que acceder una vez logueado. Implemente filtros de búsqueda para facilitar cualquier gestión que deba realizar el usuario que gestione las facultades.

Individual contributions

These are the contributions made by each member of the group.

Adriel Saa Romano

I have designed and implemented the User Interface. I have also implemented the reservations, i.e., the creation, edition and deletion of simple, periodic and grupal reservations.

I have developed the different means of reservations offered by the application: to reserve from a space or from a specific date. I've also implemented a quick search for user groups.

I have also contributed to: users' authentication, profile edition, errors control, as well as the reservations' swapping between calendars in order that to create faculty's timetables.

Jaime Mayordomo Moreno

I have designed and implemented the administrator interface. I have also implemented users, faculties, buildings and spaces' management, which include the creation, edition and deleting of each entity.

I have included, for each managed entity, one filter to improve the management in administrator mode. I've also implemented the soft-delete means, used for enabling or disabling users, faculties, buildings and spaces.

Another contribution has been including the image upload for each entity.

Javier Terrón Menoyo

I have taken care of behavior and design in manager's scope. I have worked on editing, creating, deleting and restoring users, spaces and buildings. I've also implemented reservations edition and restrictions in spaces in order to specify if those spaces' reservations required an authorization.
I contributed in a welcome feature that redirects each logged user to a different page. I worked on search filters for every management made by a faculty manager.

Apéndice A. Historias de Usuario

Historias de Usuario

En este apartado comentaremos con más detalle las distintas historias de usuario que dan una visión global de la herramienta. Éstas historias o epics se desglosan en roles para poder ver con claridad qué papel desempeña el actor.

Así pues, nos encontramos con 3 roles, ordenados de menor a mayor importancia:

- Usuario
- Gestor
- Administrador

Siguiendo el orden anterior pasamos a detallar las diferentes historias de usuario, las cuales son acumulativas, es decir, la secretaria tiene las mismas historias de usuario que un usuario, más las suyas. Lo mismo con el administrador que tiene las de un usuario, una secretaria y las suyas propias.

1. Epic e Historias de usuario: Usuario

Vamos a desarrollar la historia del usuario a muy alto nivel.

Título	*EPIC 1*
Como usuario quiero poder editar mi perfil, además quiero acceder en todo momento a la información de mis reservas.	
Descripción/Criterios de aceptación	
Debe permitir que las reservas sean únicas.	

Epic del usuario

A continuación, se desglosa el epic anterior en las diferentes historias de usuario que hemos contemplado.

Título	*U.S 1.1*
Como usuario quiero cambiar mi foto de perfil.	
Descripción/Criterios de aceptación	
Debe permitir que se puedan subir fotos al sistema.	

Cambiar foto de perfil

Título	*U.S 1.2*
Como usuario quiero cambiar mi contraseña. **Descripción/Criterios de aceptación** Debe validar que la contraseña actual es correcta.	

Cambiar contraseña

Título	*U.S 1.3*
Como usuario quiero hacer una reserva simple. **Descripción/Criterios de aceptación** Debe validar que la reserva no solape con ninguna otra.	

Crear reserva simple

Título	*U.S 1.4*
Como usuario quiero hacer una reserva periódica. **Descripción/Criterios de aceptación** Debe validar que la reserva no solape con ninguna otra.	

Crear reserva periódica

Título	*U.S 1.5*
Como usuario quiero editar una reserva simple. **Descripción/Criterios de aceptación** Debe validar los nuevos datos y comprobar que no solape con otras reservas.	

Editar reserva simple

Título	*U.S 1.6*
Como usuario quiero editar una reserva periódica. **Descripción/Criterios de aceptación** Debe validar los nuevos datos y comprobar que no solape con otras reservas.	

Editar reserva periódica

Título	*U.S 1.7*
Como usuario quiero eliminar una reserva simple. **Descripción/Criterios de aceptación** Debe comprobar que la reserva exista.	

Eliminar reserva simple

Título	*U.S 1.8*
Como usuario quiero eliminar una reserva periódica. **Descripción/Criterios de aceptación** Debe comprobar que la reserva esté dada de alta en el sistema.	

Eliminar reserva periódica

Título	*U.S 1.9*
Como usuario quiero crear un grupo de reservas. **Descripción/Criterios de aceptación** Debe comprobar que no dispongas de un grupo con ese nombre.	

Crear grupo de reservas

Título	*U.S 1.10*
Como usuario quiero editar un grupo de reservas. **Descripción/Criterios de aceptación** Debe validar los datos introducidos.	

Editar grupo de reservas

Título	*U.S 1.11*
Como usuario quiero eliminar un grupo de reservas. **Descripción/Criterios de aceptación** Debe comprobar que el grupo esté dado de alta en el sistema.	

Eliminar grupo de reservas

Título	*U.S 1.12*
Como usuario quiero pasar reservas de un grupo a otro. **Descripción/Criterios de aceptación** Debe permitir que se pueda arrastrar y soltar una reserva de un calendario a otro.	

Mover reservas entre grupos

Título	*U.S 1.13*
Como usuario quiero ver más de un grupo de reservas a la vez. **Descripción/Criterios de aceptación** Debe permitir que se pueda elegir cuántos grupos ver al mismo tiempo.	

Ver varios grupos de reservas

Título	*U.S 1.14*
Como usuario quiero ver todas las reservas de un espacio. **Descripción/Criterios de aceptación** Debe comprobar que el espacio está dado de alta en el sistema.	

Ver reservas espacio

Título	*U.S 1.15*
Como usuario quiero ver sólo las reservas de mañana de un espacio. **Descripción/Criterios de aceptación** Debe permitir seleccionar sólo las reservas de mañana.	

Ver reservas espacio de mañana

Título	*U.S 1.16*
Como usuario quiero ver sólo las reservas de tarde de un espacio. **Descripción/Criterios de aceptación** Debe permitir seleccionar sólo las reservas de tarde.	

Ver reservas espacio de tarde

Título	*U.S 1.17*
Como usuario quiero buscar qué espacios están disponibles en una fecha en concreto para reservarlo. **Descripción/Criterios de aceptación** Debe permitir seleccionar edificio, fecha de inicio y fecha de fin.	

Reservar por espacio

Título *U.S 1.18*

Como usuario quiero ver mis reservas en formato lista.

Descripción/Criterios de aceptación

Debe permitir que se muestren las reservas en formato lista.

Ver lista de reservas

Título *U.S 1.19*

Como usuario quiero ver mis reservas en formato calendario.

Descripción/Criterios de aceptación

Debe permitir que se muestren las reservas en formato calendario.

Ver calendario de reservas

Título *U.S 1.20*

Como usuario quiero ver cuántas reservas pendientes de confirmación tengo.

Descripción/Criterios de aceptación

Debe mostrar cuántas reservas pendientes tengo.

Notificación reservas pendientes

2. Epic e Historias de usuario: Gestor

Título *EPIC 2*

Como gestor quiero poder realizar cualquier gestión relacionada con las reservas de la facultad y quiero poder cambiar los requisitos para realizar una reserva. Además, quiero poder realizar cualquier operación que sea posible realizar como usuario.

Descripción/Criterios de aceptación

Debe permitir que se acceda a la información e interactuar con ella cuando sea necesario.

Epic del gestor

A continuación, se desglosa el epic anterior en las diferentes historias de usuario que hemos contemplado.

Título	*U.S 2.1*
Como gestor quiero poder realizar cualquier operación que pueda gestionar un usuario de tipo usuario (Epic 1). **Descripción/Criterios de aceptación** Debe permitir que se acceda a la información e interactuar con esta.	

Gestiones de usuario

Título	*U.S 2.2*
Como gestor quiero poder ver una lista de las reservas de la facultad de mi ámbito. **Descripción/Criterios de aceptación** Debe mostrar una lista con la información básica de las reservas.	

Ver lista reservas

Título	*U.S 2.3*
Como gestor quiero poder acceder a la información de una reserva concreta de afecte a la facultad que gestiono. **Descripción/Criterios de aceptación** Debe mostrar un cuadro de diálogo con la información de la reserva.	

Ver información de reserva

Título	*U.S 2.4*
Como gestor quiero filtrar las reservas de mi facultad por espacio. **Descripción/Criterios de aceptación** Debe existir el espacio y mostrar una lista con las reservas afectadas.	

Filtrar reservas por espacio

Título	*U.S 2.5*
Como gestor quiero filtrar las reservas de mi facultad por nombre de usuario.	
Descripción/Criterios de aceptación	
Debe existir usuario y mostrar una lista con las reservas afectadas.	

Filtrar reservas por usuario

Título	*U.S 2.6*
Como gestor quiero filtrar las reservas de mi facultad por estado de la reserva.	
Descripción/Criterios de aceptación	
Debe permitir que se muestre una lista con las reservas afectadas.	

Filtrar reservas por estado

Título	*U.S 2.7*
Como gestor quiero modificar las reservas de mi facultad.	
Descripción/Criterios de aceptación	
Debe introducirse información válida.	

Modificar reservas

Título	*U.S 2.8*
Como gestor quiero confirmar una reserva de mi facultad.	
Descripción/Criterios de aceptación	
Debe comprobar que la reserva existe	

Confirmar reserva

Título	*U.S 2.9*
Como gestor quiero denegar una reserva de mi facultad. **Descripción/Criterios de aceptación** Debe comprobar que la reserva existe	

Denegar reserva

Título	*U.S 2.10*
Como gestor quiero borrar una reserva de mi facultad. **Descripción/Criterios de aceptación** Debe comprobar que la reserva existe.	

Borrar reserva

Título	*U.S 2.11*
Como gestor quiero cambiar los criterios de aceptación de reserva para una sala que gestione mi facultad. **Descripción/Criterios de aceptación** Debe afectar a las reservas de su facultad que se creen a partir de ese momento.	

Modificar criterio de autorización.

3. Epic e Historias de usuario: Administrador

Título	*EPIC 3*
Como administrador quiero una aplicación que me permita gestionar las facultades, edificios, espacios, usuarios y reservas. **Descripción/Criterios de aceptación** Debe permitir que se acceda a la información.	

Epic del administrador

A continuación, se desglosa el epic anterior en las diferentes historias de usuario que hemos contemplado.

Título	*U.S 3.1*
Como administrador quiero ver todos los usuarios dados de alta en el sistema.	
Descripción/Criterios de aceptación	
Debe permitir que se acceda a la información.	

Ver usuarios

Título	*U.S 3.2*
Como administrador quiero buscar un usuario filtrando por nombre.	
Descripción/Criterios de aceptación	
Debe mostrar el usuario en el autocompletar.	

Buscar usuario

Título	*U.S 3.3*
Como administrador quiero ver la información de un usuario en concreto.	
Descripción/Criterios de aceptación	
Debe mostrar el usuario en el autocompletar.	

Ver la información de un usuario

Título	*U.S 3.4*
Como administrador quiero editar los datos de un usuario en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Editar un usuario

Título	*U.S 3.5*
Como administrador quiero eliminar un usuario en particular.	
Descripción/Criterios de aceptación	
Debe mostrar el usuario en el autocompletar.	

Eliminar usuario

Título	*U.S 3.6*
Como administrador quiero añadir un nuevo usuario.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Añadir nuevo usuario

Título	*U.S 3.7*
Como administrador quiero reactivar un usuario en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Reactivar usuario

Título	*U.S 3.8*
Como administrador quiero ver todas las facultades dadas de alta en el sistema.	
Descripción/Criterios de aceptación	
Debe permitir que se acceda a la información.	

Ver facultades

Título	*U.S 3.9*
Como administrador quiero ver la información de una facultad.	
Descripción/Criterios de aceptación	
Debe aparecer la facultad en el autocompletar.	

Ver facultad

Título	*U.S 3.10*
Como administrador quiero eliminar una facultad en particular.	
Descripción/Criterios de aceptación	
Debe mostrar la facultad en el autocompletar.	

Eliminar facultad

Título	*U.S 3.11*
Como administrador quiero añadir una facultad en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Añadir facultad

Título	*U.S 3.12*
Como administrador quiero reactivar una facultad en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos	

Reactivar facultad

Título	*U.S 3.13*
Como administrador quiero editar los datos de una facultad en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Editar una facultad

Título	*U.S 3.14*
Como administrador quiero buscar una facultad filtrando por su nombre.	
Descripción/Criterios de aceptación	
Debe aparecer la facultad en el autocompletar.	

Buscar facultad

Título	*U.S 3.15*
Como administrador quiero ver todos los edificios dados de alta en la aplicación.	
Descripción/Criterios de aceptación	
Debe mostrar los edificios.	

Ver todos los edificios

Título	*U.S 3.16*
Como administrador quiero buscar un edificio filtrando por el nombre.	
Descripción/Criterios de aceptación	
Debe mostrar el edificio en el autocompletar.	

Buscar edificio

Título	*U.S 3.17*
Como administrador quiero ver los datos de un edificio en particular.	
Descripción/Criterios de aceptación	
Debe mostrar los datos del edificio.	

Ver información de un edificio

Título	*U.S 3.18*
Como administrador quiero editar los datos de un edificio en particular.	
Descripción/Criterios de aceptación	
Debe validar los datos introducidos.	

Editar edificio

Título	*U.S 3.19*
Como administrador quiero eliminar un edificio en particular.	
Descripción/Criterios de aceptación	
Debe eliminar el edificio.	

Eliminar edificio

Título	*U.S 3.20*
Como administrador quiero reactivar un edificio en particular.	
Descripción/Criterios de aceptación	
Debe restaurar el edificio.	

Reactivar edificio

Título	*U.S 3.21*
Como administrador quiero ver todos los espacios que hay en la aplicación.	
Descripción/Criterios de aceptación	
Debe mostrar los espacios.	

Ver todos los espacios

Título	*U.S 3.22*
Como administrador quiero buscar un espacio filtrando por nombre.	
Descripción/Criterios de aceptación	
Debe mostrar el espacio en el autocompletar.	

Buscar espacio

Título	*U.S 3.23*
Como administrador quiero ver los datos de un espacio en particular.	
Descripción/Criterios de aceptación	
Debe mostrar los datos del espacio.	

Ver datos de un espacio

Título	*U.S 3.24*
Como administrador quiero eliminar un espacio en particular.	
Descripción/Criterios de aceptación	
Debe eliminar el espacio.	

Eliminar espacio

Título	*U.S 3.25*
Como administrador quiero reactivar un espacio en particular.	
Descripción/Criterios de aceptación	
Debe reactivar el espacio.	

Reactivar espacio

Título	*U.S 3.26*
Como administrador quiero añadir un espacio. **Descripción/Criterios de aceptación** Debe validar los datos introducidos.	

Añadir nuevo espacio

Referencias y Bibliografía

[1] http://www.aprenderaprogramar.com/index.php?option=com_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163

[2] http://aprenderaprogramar.es/index.php?option=com_content&view=article&id=435:ique-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192

[3] https://es.wikipedia.org/wiki/Thymeleaf

[4] https://es.wikipedia.org/wiki/Maven

[5] https://es.wikipedia.org/wiki/JSON

[6] Traducido de https://en.wikipedia.org/wiki/Font_Awesome

[7] https://es.wikipedia.org/wiki/Hibernate

[8] http://www.templatemonster.com/help/es/js-animated-how-to-work-with-fullcalendar-plugin.html#gref

[9] https://es.wikipedia.org/wiki/Java_Persistence_API

[10] https://cacoo.com/lang/es/

[11] http://www.desarrolloweb.com/articulos/introduccion-git-github.html

[12] https://guides.github.com/activities/hello-world/

[13] https://blog.udemy.com/git-tutorial-a-comprehensive-guide/

[14] http://docs.spring.io/spring-security/site/docs/3.0.x/reference/ns-config.html#ns-web-xml

[15] Leonard Richardson, Mike Amundsen, Sam Ruby. **RESTful Web APIs.** 2013

[16] Craig Walls. **Spring.** 2008

[17] Petri Kainulainen. **Spring Data.** 2012