



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

Aplicación web para la gestión de reservas de espacios de trabajo

Trabajo fin de estudio presentado por:	Aitor Gómez Afonso
Línea de investigación:	Diseño y desarrollo de aplicaciones. Bases de datos: almacenamiento, análisis y presentación de datos.
Director/a:	Jesús Pérez Melero
Fecha:	15-05-2023
Repositorio del código fuente:	https://github.com/aitorgomezUNIR/gestor-reservas

Resumen

Este trabajo nace de la necesidad de proporcionar al personal de secretaría una aplicación web para que puedan gestionar las reservas sobre los espacios de trabajo de su empresa. También cubre la necesidad de que este personal sea capaz de crear, editar y anular las reservas de los empleados registrados en el sistema, así como visualizar la ocupación de los espacios de trabajo para una fecha y hora concretas.

Se han definido una serie de requisitos que deberán ser cumplidos al final del desarrollo de la aplicación. Para lograr estos objetivos se ha utilizado una metodología basada en SPRINTS apoyada en la plataforma Trello, que permite crear tableros y tarjetas que representan las tareas a realizar, así como columnas que representan el estado de cada tarea.

Para el desarrollo de esta aplicación se han utilizado diferentes tecnologías como Java, Spring, PrimeFaces para la realización del código y MySQL para la realización de la base de datos.

Palabras clave: Reserva, Spring, PrimeFaces, Gestión, MVC

Abstract

This project is brought to life due to the needs of the secretarial staff of a web application that allows them to manage bookings on the company workspaces. This web application will allow the staff to create, edit, and cancel bookings of employees that are already signed up in the database. It will also let the user check the occupation of the company workspaces for a given date and time.

A number of requisites have been defined that will need to be completed at the end of the development process. To accomplish these requisites, a SPRINT based methodology has been put into practice using the Trello platform, which allows the developer to create boards and cards that represent tasks to be completed, aswell as columns that represent the state a card is currently in.

Different technologies have been used during the development process. Java and the Spring Framework have been used for the backend part, meanwhile Primefaces has been used for the front-end side of the application. MySQL has served as the database system.

Keywords: Booking, Spring, PrimeFaces, Management, MVC

Índice de contenidos

1. Introducción	9
1.1. Motivación	9
1.2. Planteamiento del trabajo	9
1.3. Estructura del trabajo	10
2. Contexto y Estado del Arte.....	11
2.1. Contexto.....	11
2.1.1. Java	12
2.1.2. Spring.....	13
2.1.3. Spring Boot	13
2.1.4. JPA (Java Persistence API)	14
2.1.5. Spring Data JPA.....	14
2.1.6. PrimeFaces.....	15
2.1.7. MySQL.....	16
2.1.8. Trello	17
2.2. Estado del arte	18
2.2.1. Calendario de Outlook.....	18
2.2.2. Calendario de Google (Google Calendar)	20
2.2.3. Kalena	21
2.2.4. Condeco	27
3. Objetivos y metodología de trabajo.....	29
3.1. Objetivo general.....	29
3.2. Objetivos específicos	29
3.3. Metodología de trabajo	29
4. Estructura del proyecto.....	31

4.1.	Casos de uso.....	32
4.1.1.	Iniciar sesión	32
4.1.2.	Crear reserva de puesto de trabajo.....	35
4.1.3.	Crear reserva de sala de reuniones	38
4.2.	Base de datos	40
4.2.1.	Diagrama entidad-relación	40
4.2.2.	Especificación de las tablas	42
4.3.	Front-end y back-end.....	49
4.3.1.	Diseño de las vistas de la aplicación.....	51
4.3.2.	Login (Inicio de sesión) y Logout	53
4.3.3.	Búsqueda	59
4.3.4.	Listado de recursos.....	62
4.3.5.	Creación de reserva de puesto de trabajo	69
4.3.6.	Ficha de reserva de puesto de trabajo	74
4.3.7.	Edición de reserva de puesto de trabajo.....	76
4.3.8.	Creación de reserva de sala de reuniones.....	78
4.3.9.	Ficha de reserva de sala de reuniones	82
4.3.10.	Edición de reserva de sala de reuniones	85
4.4.	Evaluación de resultados	88
5.	Conclusiones y trabajo futuro	90
5.1.	Conclusiones del trabajo.....	90
5.2.	Líneas de trabajo futuro	94
	Referencias bibliográficas.....	95
	Índice de acrónimos	96

Índice de figuras

Figura 1. Creación de evento mediante el calendario de Outlook, agregando una sala de reuniones como invitado.....	19
Figura 2. Creación de evento mediante el calendario de Google, agregando una sala de reuniones como invitado.....	20
Figura 3. Visualización del plano y su ocupación en la funcionalidad Reserva desde plano de Kalena.	23
Figura 4. Creación de una reserva en la funcionalidad Reserva desde plano de Kalena.	24
Figura 5. Edición de una reserva en la funcionalidad Reserva desde plano de Kalena.	25
Figura 6. Visualización del calendario dentro de la funcionalidad Reserva desde calendario de Kalena.	26
Figura 7. Visualización del calendario de la aplicación de Escritorio de Condeco.	27
Figura 8. Visualización del plano interactivo de la aplicación de Escritorio de Condeco.....	28
Figura 9. Diagrama de secuencia de inicio de sesión.	34
Figura 10. Diagrama de secuencia del caso de uso "Crear reserva de puesto de trabajo".	37
Figura 11. Diagrama entidad-relación de la base de datos.	41
Figura 12. Diagrama de arquitectura de la aplicación.....	50
Figura 13. Estructura de la interfaz en base a la plantilla creada para la aplicación.	52
Figura 14. Vista de inicio de sesión.....	56
Figura 15. Vista de error de inicio de sesión.	57
Figura 16. Vista de logout.....	58
Figura 17. Vista de búsqueda.	61
Figura 18. Vista de listado de recursos.....	66
Figura 19. Panel lateral mostrar filtros.....	67
Figura 20. Panel lateral de un recurso específico.....	68

Figura 21. Creación de reserva de puesto de trabajo.	71
Figura 22. Búsqueda de organizador en creación de reserva de puesto de trabajo.	72
Figura 23. Diálogo de conflictividad al crear reserva de puesto de trabajo.	73
Figura 24. Ficha de reserva de puesto de trabajo.	75
Figura 25. Edición de reserva de puesto de trabajo.....	77
Figura 26. Creación de reserva de sala de reuniones.....	80
Figura 27. Diálogo de conflictividad al crear reserva de sala de reuniones.....	81
Figura 28. Ficha de reserva de sala de reuniones.	83
Figura 29. Opciones sobre la reserva en ficha de reserva de sala de reuniones.	84
Figura 30. Edición de reserva de sala de reuniones.	87
Figura 31. Vista del listado de recursos a una resolución de 1600x900.	89
Figura 32. Resultados del formulario de usabilidad	91
Figura 33. Tablero de Trello al acabar el desarrollo de la aplicación.	93

Índice de tablas

Tabla 1. Caso de uso "Iniciar sesión".....	32
Tabla 2. Caso de uso "Crear reserva de puesto de trabajo"	35
Tabla 3. Caso de uso "Crear reserva de sala de reuniones".....	38

1. Introducción

En este capítulo se explica el motivo por que se ha elegido desarrollar esta aplicación, así como a qué necesidad responde su creación. También se explica cómo se ha planteado el trabajo para dar cubrir la necesidad y por último, que se puede esperar encontrar el lector de esta memoria en cada capítulo del trabajo.

1.1. Motivación

La empresa para la que trabajo, Bookker, es una empresa dedicada a la gestión de recursos de otras empresas (usuarios y espacios de trabajo). Esta gestión de recursos involucra la gestión de reservas desde las aplicaciones móviles por parte de cada empleado, así como la gestión de los propios recursos de la compañía desde un panel específico de gestión. Gracias al ecosistema proporcionado por Bookker, las compañías que adquieren el servicio logran ahorrar una gran cantidad de tiempo y dinero en dicha gestión.

El desarrollo de este trabajo nace de la necesidad de otorgar al personal de secretaría de una compañía la capacidad de gestionar (crear, editar y anular) reservas sobre el resto de los empleados de la compañía, y de esta forma, crear reservas a terceros.

Además de poder gestionar las reservas, el personal de secretaría necesita saber la ocupación de los recursos en todo momento para así saber que recursos están libres para ser reservados y cuáles no.

1.2. Planteamiento del trabajo

Para cubrir esta necesidad por parte del personal de secretaría, se plantea el desarrollo de una aplicación web donde dicho personal pueda iniciar sesión y gestionar reservas a empleados de su compañía, ofreciendo la posibilidad de ver la ocupación de los recursos en todo momento y para el horario y fecha que el usuario desee.

Desde esta aplicación web, aparte de ver la ocupación de los recursos, los usuarios deben poder crear, editar y anular reservas sobre los recursos (puestos de trabajo y salas de reuniones) de la compañía.

A la hora de elegir las tecnologías a usar para el desarrollo de la aplicación, se ha elegido utilizar Java como lenguaje de programación, Spring Framework como añadido sobre Java y PrimeFaces para la parte dedicada al front-end de la aplicación.

1.3. Estructura del trabajo

Esta memoria esta dividida en varios capítulos que abordan diferentes temas relacionados con el trabajo desarrollado, el lector puede esperar lo siguiente en cada capítulo:

- **Contexto y estado del arte:** En este capítulo se explica la problemática abordada con este trabajo y estado del mercado actual respecto a soluciones y/o productos centrados solucionar necesidades como la identificada en este trabajo. También se explican las tecnologías que se usaron para el desarrollo de la aplicación.
- **Objetivos y metodología de trabajo:** Capítulo dedicado a definir los objetivos que se esperan conseguir una vez se complete el desarrollo de la aplicación, así como la metodología usada durante el proceso de desarrollo, y que ayudará en la finalización de éste.
- **Estructura del proyecto:** El capítulo mas extenso de la memoria, donde se entra en detalle en el contenido de la aplicación, incluyendo la especificación completa de la base de datos, así como la explicación del funcionamiento de cada vista de la aplicación, aportando imágenes para que el lector pueda visualizar cada vista. En este capítulo también está presente un apartado dedicado a la evaluación de resultados, donde se aborda si se han cumplido los resultados esperados de la aplicación.
- **Conclusiones y trabajo futuro:** En este capítulo se plasma el resumen final del trabajo, si se han cumplido todos los objetivos propuestos inicialmente, así como las posibles líneas de trabajo futuro que podrían mejorar la aplicación.

2. Contexto y Estado del Arte

En este capítulo se presenta el contexto de la aplicación, así como el porqué de la elección de las tecnologías utilizadas. También se pone en situación el estado del arte en cuanto a las soluciones existentes y que hacen competencia de una u otra forma a la solución desarrollada en este proyecto.

2.1. Contexto

El desarrollo de esta aplicación nace de la necesidad por parte de medianas y grandes empresas de gestionar sus recursos (salas de reuniones, puestos de trabajo y plazas de parking) para así ahorrar costes.

Hasta el momento, dichas empresas utilizan sistemas analógicos o bien sistemas digitales de poca fiabilidad para poder decidir, por ejemplo, qué puesto de trabajo pertenece a cada trabajador, controlar su hora de llegada y/o salida (que nuestra solución permite controlar mediante un sistema de check-in y check-out) o, en el caso de salas de reuniones, gestionar su uso y disponibilidad.

El objetivo de la aplicación es permitir a las personas encargadas de la gestión de los recursos de la empresa (sus salas de reuniones y puestos de trabajo) utilizar un sistema que les asista en la creación, edición y cancelación de reservas sobre los diferentes recursos para los empleados de la compañía.

Este tipo de aplicaciones surgen de la necesidad de las empresas de ahorrar costes en la gestión manual de los espacios de trabajo, ya que se delega esta gestión a la lógica de la aplicación y al uso que le dan los empleados, y por tanto, ahorrando una cantidad de dinero notable a la empresa ya que no se tiene que preocupar de primera mano de esta gestión.

Además, a raíz de la pandemia, y la incorporación del modelo de trabajo híbrido (teletrabajo junto con trabajo presencial en oficina) ha ocasionado que los espacios de trabajo dejen de ser fijos para una persona y se conviertan en espacios compartidos por múltiples trabajadores, lo que dificulta la gestión de los espacios de trabajo si hay que realizarla manualmente, en vez de que una aplicación inteligente se encargue de ello.

La aplicación web desarrollada en este proyecto complementa las aplicaciones móviles ya existentes en el ecosistema de Bookker, y desde las que los empleados, individualmente, pueden consultar, editar y cancelar sus reservas, así como crear nuevas.

Las características principales que hacen única a nuestra aplicación son la capacidad de confirmar la asistencia de un empleado, así como poder visualizar la ocupación de los recursos de la compañía para una fecha y hora concreta.

También será posible consultar las reservas de cada recurso y realizar acciones sobre éstas, como por ejemplo editarlas y/o cancelarlas.

Las tecnologías utilizadas en el desarrollo de esta aplicación han sido escogidas tanto por su popularidad como por su robustez y establecimiento en el mundo del desarrollo. Java, por ejemplo, es un lenguaje de programación que lleva presente en el campo del desarrollo software mucho tiempo, lo que ofrece una gran de documentación y soporte en caso de que exista algún tipo de problema. Además, y gracias a esto, a lo largo del tiempo se han creado y pulido herramientas o frameworks que se apoyan en Java para su funcionamiento. Uno de estos frameworks es Spring, y su serie de librerías que nos ayudan en varios aspectos y nos permiten abstraernos de tareas, que, sin este framework, deberíamos implementar desde cero.

Para la parte del front-end, se ha escogido utilizar PrimeFaces ya que, igual que Java y Spring, lleva años en desarrollo y eso ha permitido a sus creadores pulirlo enormemente, ofreciendo a los desarrolladores que deciden usarlo una gran robustez en su funcionamiento, además de gran facilidad de uso.

A continuación, se detallan las tecnologías que se han utilizado en este proyecto:

2.1.1. Java

Java es un lenguaje de programación muy extendido, y, por tanto, dispone de diferentes *frameworks* (entornos de trabajo que gracias a sus herramientas permiten solucionar problemas de forma sencilla) para ayudarnos a conseguir nuestro objetivo.

Uno de estos *frameworks*, cuyo nombre es Spring, es el encargado de ayudarnos en el desarrollo de la aplicación.

La versión de Java que se utiliza es Java 11, que fue lanzada hace unos años y se ha establecido como la versión principal de Java después de Java 8, siendo esta última la principal por todo el software de legado presente en el mercado. (Adams et al., 2023)

2.1.2. Spring

Spring es el *framework* que utilizado en el desarrollo de la aplicación. Fue lanzado en 2003 y desde entonces se ha establecido como uno de los frameworks de Java más populares. La finalidad de Spring es (Yanina Muradas, 2018) “estandarizar, agilizar, manejar y resolver los problemas que puedan ir surgiendo en el trayecto de la programación”.

Spring nos permite abstraernos en gran medida de diferentes aspectos, como, por ejemplo:

- La conexión y la realización de operaciones CRUD (Crear, Leer, Actualizar y Borrar) sobre la base de datos y sus diferentes tablas.
- El despliegue de la aplicación en una máquina para poder uso de la aplicación. No es necesario de un servidor web ya que Spring posee uno integrado.

En los siguientes apartados se explicarán en detalle las tecnologías concretas de Spring que se han utilizado en el desarrollo de la aplicación.

2.1.3. Spring Boot

Spring Boot es una de las tecnologías presentes en el ecosistema del framework de Spring que nos asistió durante el desarrollo.

El objetivo principal de Spring Boot es ayudar a crear aplicaciones *stand-alone* que se pueden ejecutar con el mínimo esfuerzo por parte del desarrollador. (Spring Framework, s. f.)

Para cumplir con el objetivo de liberar al desarrollador, Spring Boot cuenta con las siguientes funcionalidades:

- Tomcat embebido, que nos permite correr la aplicación y que Spring se encargue de generar un servidor propio. De esta forma nos evita tener que configurar un servidor propio donde alojar la aplicación.
- Sugerir dependencias que puedan ser útiles a la hora de crear el proyecto, evitando que tengamos que perder el tiempo buscándolas más adelante.

- Capacidad de configurar Spring y librerías de terceros siempre que sea posible. Esto permite que, al añadir una librería externa en el archivo de configuración del proyecto, Spring Boot se encargue de gestionar las dependencias de las librerías por nosotros.

2.1.4. JPA (Java Persistence API)

JPA es una API de persistencia que permite manejar datos relaciones, consiguiendo mantener las ventajas de la programación orientada a objetos al interactuar con una base de datos.

Esto se consigue mediante la creación de entidades, que son clases Java ligeras cuyo objetivo es representar las tablas de la base de datos. Mediante anotaciones Java, podemos asociar cada atributo de la clase Java a la columna correspondiente de la tabla dentro de la base de datos. (Alarcón, 2021)

Cada instancia de cada entidad se corresponde a un registro de la tabla que dicha entidad representa.

También es posible relacionar diferentes entidades, para así cumplir con las diferentes relaciones que puede existir en una base de datos relacionas (claves foráneas o *foreign keys*, etc.).

2.1.5. Spring Data JPA

Spring DATA JPA nos permite implementar repositorios basados en JPA (Java Persistence API) que se usarán para gestionar operaciones CRUD contra la base de datos. (Spring, s. f.)

Para lograr esto, deberemos tener entidades (clases Java) en nuestro proyecto que representen las diferentes tablas de la base de datos, y un repositorio para cada entidad. Este repositorio es el que contiene diferentes funciones que, siguiendo la sintaxis proporcionada por JPA o usando una consulta nativa, nos permite realizar acciones CRUD contra la base de datos.

Este proceso de crear entidades que representen las tablas de la base de datos se conoce como *mapping* (mapear).

Por ejemplo, si en nuestra tabla tenemos una tabla llamada *usuario*, en nuestro proyecto deberemos tener una clase Java llamada Usuario que contenga los mismos campos que la tabla en base de datos. Si queremos crear un usuario nuevo, deberemos crear un nuevo objeto Usuario y solicitar al repositorio que guarde dicho objeto mediante el método *save*.

Internamente, JPA realiza la consulta INSERT correspondiente, creando en este caso el nuevo registro de usuario en la tabla de base de datos.

Usando Spring Data JPA, conseguimos trabajar con la base de datos directamente desde la aplicación, sin tener que preocuparnos (normalmente) del uso de consultas nativas SQL.

2.1.6. PrimeFaces

PrimeFaces es una librería de componentes que nos ayudará en el diseño y creación de las diferentes vistas en nuestra aplicación, o, dicho de otra forma, la interfaz de usuario.

Debido a que los componentes de PrimeFaces están creados para el framework JavaServer Faces (JSF), podemos utilizar las herramientas y tecnologías que este último incluye (Wikipedia Authors, s. f.):

- Conjunto de APIs para manejar eventos ocurridos del lado cliente, así como del lado del servidor, gestionar el estado de los componentes de la página web y definir un esquema de navegación de las páginas.
- Un conjunto de componentes por defecto (aunque en nuestro caso usaremos los proporcionados por PrimeFaces).
- *Beans* administrados, que nos servirán como controladores y como enlace con la interfaz de usuario.

El método de funcionamiento de PrimeFaces consiste en tener diferentes archivos XHTML que funcionan como la “vista”, donde se establece una estructura muy parecida a un documento HTML tradicional, pero utilizando las diferentes etiquetas y componentes proporcionados por PrimeFaces.

También es posible utilizar etiquetas nativas de HTML 5 en caso de que sea necesario, pero estas etiquetas no cuentan con las funcionalidades añadidas que proporcionan los componentes específicos de PrimeFaces.

La versión utilizada de PrimeFaces será la 10.0, ya que es una de las últimas versiones estables publicadas.

Aunque utilizar una versión relativamente reciente o incluso actualizar la versión en una aplicación ya existente y que lleve cierto tiempo establecida puede ser un proceso relativamente costoso e incluso peligroso (debido a que una nueva versión puede suponer

que el comportamiento de diferentes componentes haya sido alterado, ocasionando problemas de funcionamiento en la aplicación existente), tenemos el lujo de poder utilizar la versión más reciente ya que estamos desarrollando una aplicación completamente desde cero.

Desde la propia página de PrimeFaces, y en concreto dentro de su propio showcase, podemos observar todos los componentes disponibles, así como acceder a la documentación por parte de servidor (la que usaremos dentro de nuestra aplicación en los archivos XHTML) así como a la documentación para la parte de cliente (más orientada en el JavaScript de cada componente). (PrimeTek, s. f.-a)

2.1.7. MySQL

MySQL es el gestor de bases de datos relacional que se ha usado en el desarrollo de aplicación. Debido a su facilidad de uso y su gran potencial, cubre con creces las necesidades de nuestra aplicación.

MySQL Server es el encargado de permitirnos tener una base de datos en un entorno local, mientras que MySQL Workbench nos asiste en la gestión propia de la base de datos, donde se encuentran las diferentes tablas y desde donde podemos manejar su información.

Desde el editor visual de MySQL Workbench podemos acceder a las siguientes funcionalidades destacables (MySQL, s. f.):

- Editor de tablas: Nos permite gestionar las tablas dentro de nuestra base de datos
- Ventana de resultados: Nos permite visualizar los resultados de las consultas realizadas.
- Exportación de resultados: Nos permite exportar resultados de consultas, así como también exportar tablas o bases de datos enteras.
- Panel de búsqueda, para buscar las diferentes tablas de una base de datos, esta funcionalidad es especialmente útil en bases de datos con un número de tablas considerables.

La versión que se va a utilizar de MYSQL es la versión 8, que, aunque ya tiene unos años, es una versión muy estable y que ofrece las funcionalidades suficientes para el desarrollo de la aplicación.

2.1.8. Trello

Trello ayudó en la gestión de las tareas a realizar durante el desarrollo. Su facilidad de uso e interfaz amigable mediante el uso de tarjeta hace que el seguimiento de las tareas llevadero.

Además, permite incluir límites de tiempo, comentarios y/o adjuntar archivos en las tareas, así como establecer mediante la creación de columnas personalizadas gracias a su flexibilidad. (Natchimuthu, 2022)

Mediante la creación de diferentes columnas, podemos definir los diferentes estados en los que una tarjeta (tarea) se puede encontrar, como, por ejemplo:

- Sin empezar
- Activa
- Terminada (DONE)

De esta forma nos fue mucho más fácil organizar las diferentes tareas a realizar, así como conocer en todo momento el estado de cada tarea.

2.2. Estado del arte

Actualmente existen varias herramientas de gestión de recursos que sirven de competencia a nuestra aplicación.

Se definen los siguientes criterios sobre los que se valoran el resto de las aplicaciones:

- Cantidad de espacios de trabajo que es capaz de gestionar.
- Necesidad de hardware para el funcionamiento de la solución.
- Parámetros por los que se pueden filtrar los diferentes tipos de espacios de trabajo.
- Capacidad para visualizar la ocupación de los espacios de trabajo para una fecha y hora concretas.
- Capacidad de realizar check-in y check-out (confirmación de asistencia).

2.2.1. Calendario de Outlook

El calendario de Outlook es una herramienta software con la capacidad de gestionar eventos (crear, editar y eliminar) dentro del calendario de cada usuario y de añadir una sala de reuniones a estos eventos como invitado, bloqueando esa sala de reuniones al resto de usuarios que la intenten reservar. De esta forma se “reserva” esa sala de reuniones para el creador del evento en el calendario y los invitados que el creador haya añadido.

El calendario de Outlook sólo permite gestionar salas de reuniones, por lo que no es posible gestionar otros tipos de espacios de trabajo que una compañía pueda poseer en sus instalaciones.

Es posible visualizar los diferentes eventos de las salas de reuniones activando el calendario de la sala en concreto, pero esto es un proceso tedioso para el usuario ya que para obtener una visión global de la ocupación debería activar todos los calendarios de todas las salas de reuniones de la organización, siendo esto algo inviable en cuanto a usabilidad.

Hasta la fecha, Outlook no ha implementado ningún sistema de confirmación de asistencia (check in y check out).

Figura 1. Creación de evento mediante el calendario de Outlook, agregando una sala de reuniones como invitado.

▶ Enviar | ▼

● Calendario ▼

👤 +2

Agregar un título

👤

Sala 1 Blues ✕

Opcional

Horas sugeridas

Preferencias

Mar 30/05

8:00 - 8:30

✓ Disponible: todos los usuarios

Mar 30/05

8:30 - 9:00

✓ Disponible: todos los usuarios

Mar 30/05

9:00 - 9:30

✓ Disponible: todos los usuarios

🕒

14/05/2023

📅

8:00

▼

🔴

Todo el día

🌐

Zonas horarias

14/05/2023

📅

8:30

▼

🔄

No repetir ▼

📍

Buscar una sala o ubicación

🔴

Reunión de Teams

📄

Agregar una descripción o adjuntar documentos

📎 ▼

🖼️

😊

🔗

☀️

✍️

🕒

📎

2.2.2. Calendario de Google (Google Calendar)

El calendario de Google ofrece un sistema bastante parecido al calendario de Outlook, sin diferencias notables, ya que igual que Outlook, solo permite gestionar salas de reuniones, no hay forma de visualizar la ocupación de forma rápida y amigable visualmente hablando para el usuario, y, por último, tampoco ha implementado un sistema de confirmación de asistencia. (Google, s. f.)

Figura 2. Creación de evento mediante el calendario de Google, agregando una sala de reuniones como invitado.

× Añade un título Guardar

29 de may de 2023 5:00pm a 6:00pm 29 de may de 2023 Zona horaria

☐ Todo el día No se repite

Detalles del evento Encontrar un hueco

Unirme con Google Meet

Añade una ubicación

Notificación 10 minutos ×

Añadir una notificación

No disponible Visibilidad predeterminada ?

La disponibilidad podría mostrarse en otras aplicaciones de Google ?

Crear notas de la reunión

Añade una descripción

Invitados Salas

Buscar sala o recurso

Edificio prueba 1 1 🗨 🔗 ⌵

☒ Sala prueba 1 planta 1 👤 8

2.2.3. Kalena

Kalena permite gestionar reservas de cualquier espacio de trabajo o recurso corporativo de una empresa, como por ejemplo salas de reuniones, plazas de aparcamiento, puestos de trabajo, comedor, despachos y taquillas desde una misma interfaz. (Kalena, s. f.)

Igual que nuestra solución, no necesita de ningún tipo hardware específico para la gestión de los espacios de trabajo, lo cual da muchos puntos extra a la solución y la convierte instantáneamente en una opción a valorar para muchas empresas que buscan una solución de este tipo (debido al ahorro que supone no necesitar adquirir y mantener dicho hardware).

Kalena destaca las siguientes funcionalidades principales:

2.2.3.1. Reserva desde plano

Permite visualizar, desde un plano interactivo, la ocupación de todos los espacios de trabajo disponibles en la planta de un edificio concreto y para una fecha y hora determinada. La ocupación de los espacios de trabajo se define mediante una leyenda de colores.

Además, es posible pulsar en cada espacio de trabajo para crear una reserva, pudiendo elegir diferentes parámetros como el organizador, el asunto de la reserva, la hora de inicio y fin, una descripción y si se desea recurrencia (es decir, si se quiere que esta reserva se repita en el futuro).

También es posible editar una reserva ya existente. En el formulario de edición se pueden modificar los mismos datos que fueron introducidos en el formulario de creación.

2.2.3.2. Reserva desde calendario

Kalena ofrece también la posibilidad de crear y modificar reservas desde una interfaz de calendario, parecido a lo ofrecido por Google Calendar o el calendario de Outlook.

La diferencia principal con los calendarios de Google y de Outlook es que no es posible visualizar la ocupación o los eventos de cada sala para múltiples fechas de forma simultánea.

Esta funcionalidad permite filtrar los diferentes espacios de trabajo en base a varios parámetros, siendo estos el nombre, la localización y la capacidad.

De la misma forma que con la funcionalidad de Reserva desde plano, es posible crear y editar reservas desde esta interfaz.

La diferencia notable entre la solución de Kalena y nuestra aplicación es la opción de Reservar desde un plano, así como la gran cantidad de tipos de espacios de trabajo que esta solución es capaz de gestionar.

Como puntos a favor para nuestra aplicación, tendremos una capacidad de filtrado de espacios de trabajo mucho más avanzada, así como una visualización de la ocupación más precisa y extensa (no sólo decir si está ocupado o no, si no si está libre, ocupado totalmente u ocupado parcialmente).

Por lo demás, Kalena es, a primera vista, una solución bastante completa y un competidor importante a nuestra aplicación.

Figura 3. Visualización del plano y su ocupación en la funcionalidad Reserva desde plano de Kalena.



Figura 4. Creación de una reserva en la funcionalidad Reserva desde plano de Kalena.

The screenshot displays the KALENA web application interface. A modal window titled "Reserva de Puesto 001" is centered over a floor plan background. The modal contains the following fields:

- Organizador:** A dropdown menu with "juan.perez" selected.
- Asunto:** A text input field containing "Asistencia presencial".
- Hora inicio:** A time selection field set to "15:00".
- Hora fin:** A time selection field set to "18:00".
- Comentarios:** A text area containing "labores de diseño".
- Recurrencia:** A dropdown menu with "No" selected.

Below the form fields is a "Guardar" (Save) button. In the background, the floor plan shows various workspaces. To the left of the modal, there is a sidebar with icons for different workspace types and a "Localización" dropdown set to "Puestos_Panel". To the right, there is a "Hora fin" dropdown set to "18:00" and a user profile icon with the text "Hola, Juan".

Figura 5. Edición de una reserva en la funcionalidad Reserva desde plano de Kalena.

The screenshot displays the KALENA web application interface. A modal window titled "Reserva de Puesto 001" is open, allowing for the editing of a reservation. The modal contains the following fields:

- Organizador:** A dropdown menu with "juan.perez" selected.
- Asunto:** A text input field containing "Asistencia presencial".
- Hora inicio:** A time selection field set to "15:00".
- Hora fin:** A time selection field set to "18:00".
- Comentarios:** A text area containing "labores de diseño".
- Recurrencia:** A dropdown menu with "No" selected.

At the bottom of the modal are two buttons: "Modificar" (light blue) and "Eliminar" (dark blue). A mouse cursor is positioned over the "Modificar" button. In the top right corner of the modal, there is a "Ver recurso" link and a close button (X).

The background shows a floor plan of a building with various rooms and desks. A sidebar on the left contains icons for different workspace types. The top of the application has the "KALENA" logo and a user profile section showing "Hola, Juan".

Figura 6. Visualización del calendario dentro de la funcionalidad Reserva desde calendario de Kalena.

The screenshot displays the KALENA web application interface. At the top, the header shows the KALENA logo on the left and a user profile with the name 'Hola, Juan' on the right. Below the header, there is a sidebar with several icons representing different reservation categories. The main content area features a filter section with the following elements:

- Está en salas...**: A dropdown menu.
- Seleccione Fecha**: A date picker showing '13/01/2021'.
- Seleccione Sala**: A dropdown menu showing 'Todas'.
- Nombre**: A text input field with 'Ej: Sala Juntas'.
- Localización**: A text input field with 'Ej: Módulo A'.
- Capacidad**: A text input field with '10'.
- Buttons for 'X' (close) and 'Q' (search).

Below the filters, the calendar grid is displayed. The grid has two columns: 'Sala S05 Code' and 'Sala S08 Blo5'. The rows represent time slots from 08:00 to 20:30 in 30-minute increments. The grid shows reservations for the selected date and filters. The 'Sala S05 Code' column has reservations for 08:00-10:30, 12:00-13:30, and 14:00-15:30. The 'Sala S08 Blo5' column has a reservation for 14:00-15:30. The reservation for 14:00-15:30 in 'Sala S08 Blo5' is highlighted in green.

2.2.4. Condeco

Condeco es una solución que permite gestionar los espacios de trabajo de una empresa, en concreto los puestos de trabajo y salas de reuniones.

Permite utilizar software (tanto de escritorio como aplicación móvil) en los que se permite la creación de reservas sobre puestos de trabajo mediante un plano interactivo (parecido a la solución de Kalena) así como desde un calendario. (Condeco, s. f.)

Para la gestión de salas de reuniones, es necesario usar hardware en forma de tablets para gestionar las reservas de cada sala de reunión, no siendo posible, a primera vista, gestionar las salas de reuniones desde la aplicación de escritorio que ofrece Condeco.

Esta solución, igual que la nuestra, ofrece confirmación de asistencia (check-in y check-out).

La aplicación de escritorio de Condeco también es capaz de mostrar la ocupación de los puestos de trabajo de la misma forma que hace Kalena, indicando si está libre o no.

Figura 7. Visualización del calendario de la aplicación de Escritorio de Condeco.

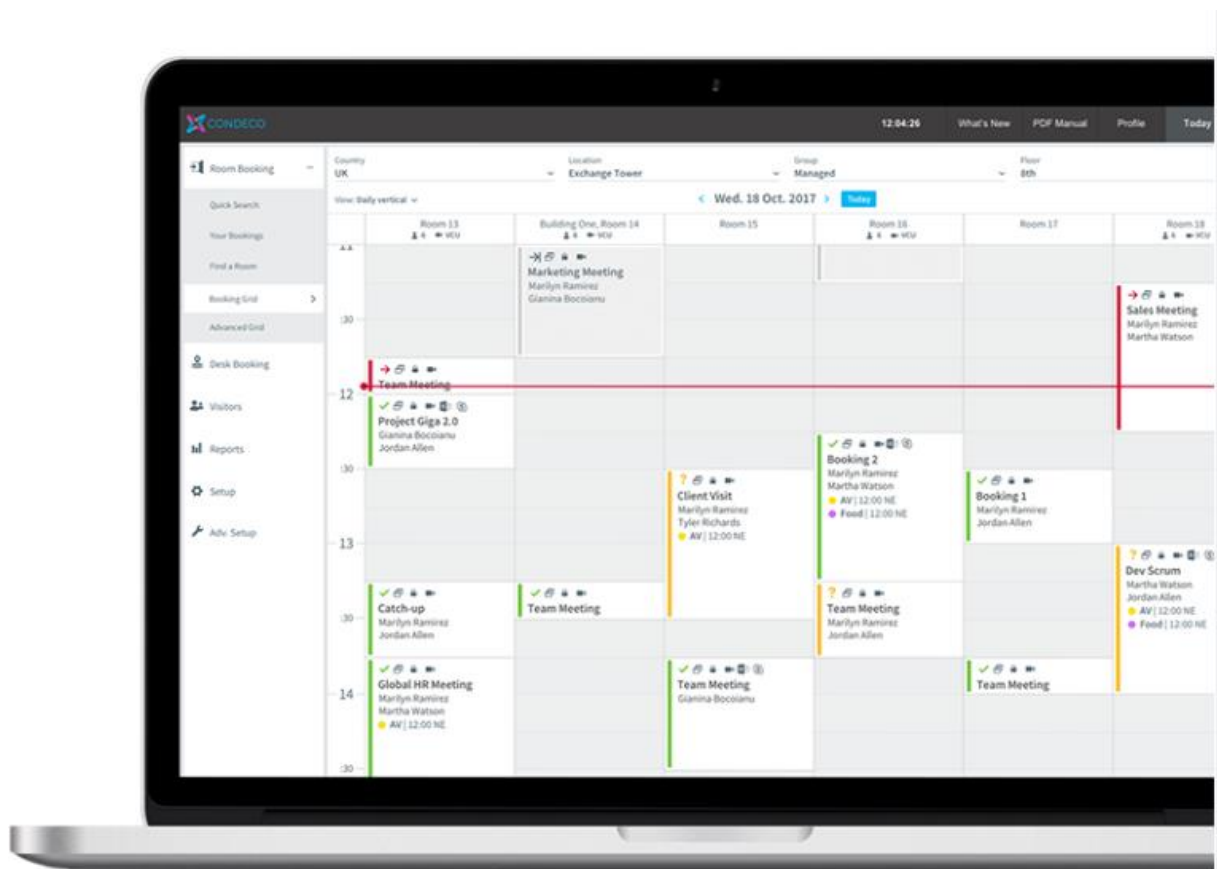
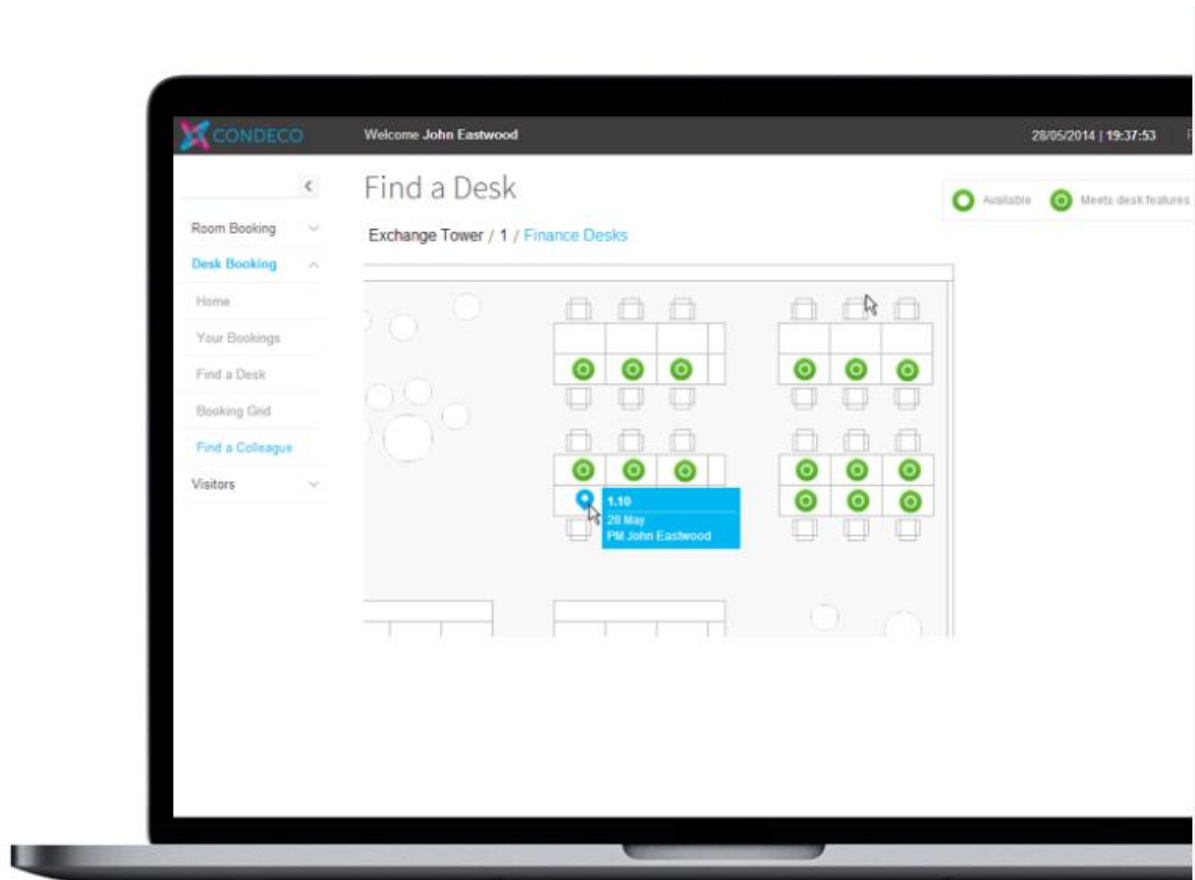


Figura 8. Visualización del plano interactivo de la aplicación de Escritorio de Condeco.



3. Objetivos y metodología de trabajo

En este capítulo se indica tanto el objetivo general de la aplicación como los objetivos que se desean cumplir una vez acabado el desarrollo. Estos requisitos serán tanto funcionales como de negocio.

Además, se comentará la metodología de trabajo escogida y que acompañará en la duración total del desarrollo de la aplicación.

3.1. Objetivo general

Ayudar al personal de secretaría a gestionar las reservas de sus empleados sobre los recursos de la empresa, ahorrando tiempo y costes de gestión.

3.2. Objetivos específicos

A continuación, se listan los hitos a conseguir durante el desarrollo:

- Implementación de una aplicación web que permita gestionar reservas.
- Capacidad para realizar la confirmación de asistencia en nombre de un empleado (check-in).
- Poder observar los recursos y su ocupación para una fecha y hora concretas.
- Conseguir que la usabilidad de la aplicación sea la correcta y esperada, evitando confusiones que puedan surgir durante el uso de la aplicación por parte de los usuarios finales.
- Llevar a cabo una planificación ágil durante el desarrollo de la aplicación.

3.3. Metodología de trabajo

Para mantener un control de las tareas a realizar, se ha utilizado la herramienta Trello y un flujo de trabajo lo más parecido a una metodología ágil SCRUM con Sprints de una semana de duración, donde se establecen una serie de tareas y funcionalidades a terminar antes del fin de cada Sprint.

Naturalmente, al no ser un proyecto que involucra a diferentes personas, así como a un SCRUM Master, es imposible ser fiel a la metodología ágil SCRUM, pero se usó como guía durante el desarrollo de la aplicación.

Usando Trello, podemos crear diferentes columnas que albergan tarjetas. Estas columnas establecen en qué estado se encuentra cada tarjeta en base a la columna en la que dicha tarjeta este situada.

Por ejemplo, si definimos una columna llamada “Done (Hecho)”, al colocar una tarjeta dentro de dicha columna, significará que esa tarjeta o tarea ha sido completada.

Gracias a esto, podemos definir una serie de tareas al comienzo de cada Sprint, y, mediante las columnas, saber en qué estado se encuentra cada tarea en cada momento.

Una vez el desarrollo de la aplicación se haya completado, todas las tareas previamente creadas deberán estar en dicha columna de “Done”.

4. Estructura del proyecto

En este capítulo se explica en detalle la estructura del proyecto, desde el nivel más bajo, la base de datos, hasta el nivel más alto, la interfaz de usuario.

Antes de entrar en la especificación de la base de datos y posteriormente del front-end y back-end, se plasman diferentes casos de uso identificados durante el planteamiento de la aplicación web.

Primero se explica la estructura de la base de datos, plasmando las tablas presentes en ésta, sus diferentes campos y las relaciones entre ellas.

Una vez terminado el capítulo asociado a la base de datos, se explica de forma extensa el funcionamiento de cada diferente vista presente en la aplicación, desde el funcionamiento de la interfaz a la lógica interna, proporcionando capturas de pantalla allí donde sea necesario para proporcionar una imagen visual de la aplicación.

4.1. Casos de uso

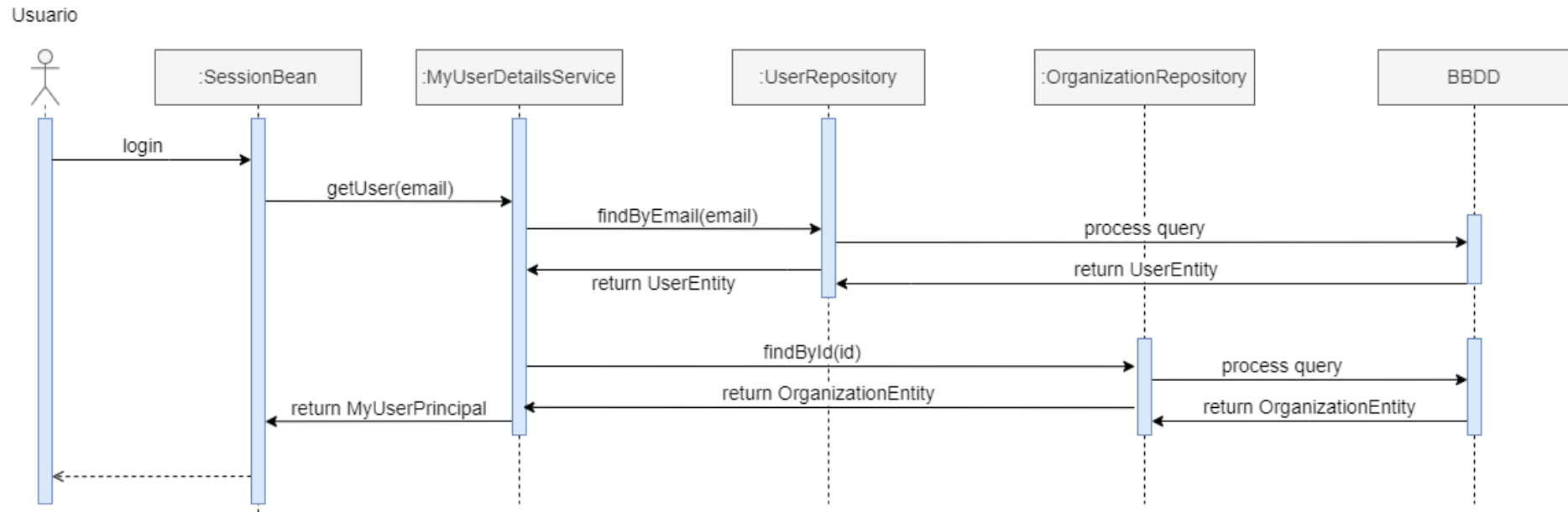
4.1.1. Iniciar sesión

Tabla 1. Caso de uso "Iniciar sesión".

ID y nombre	UC-01. Iniciar sesión
Descripción	El usuario tiene que ser capaz de iniciar sesión en la aplicación.
Disparador	El usuario carga la aplicación web sin tener la sesión iniciada.
Actor	Usuario
Resultado	El usuario ha iniciado sesión
Escenario principal	<ol style="list-style-type: none">1. El usuario introduce su email en el campo de texto dedicado al email.2. El usuario introduce su contraseña en el campo de texto dedicado a la contraseña.3. El usuario pulsa en el botón "Iniciar sesión".4. El sistema coteja el email introducido por el usuario con el listado de usuarios presentes en la base de datos.5. El sistema coteja la contraseña del usuario encontrado en la base de datos con la introducida por el usuario.6. El sistema autentica al usuario y le concede acceso a la aplicación web

Excepciones	El usuario no ha introducido un email, contraseña o bien los datos introducidos son incorrectos: El sistema lanza un mensaje de error de inicio de sesión.
--------------------	--

Figura 9. Diagrama de secuencia de inicio de sesión.



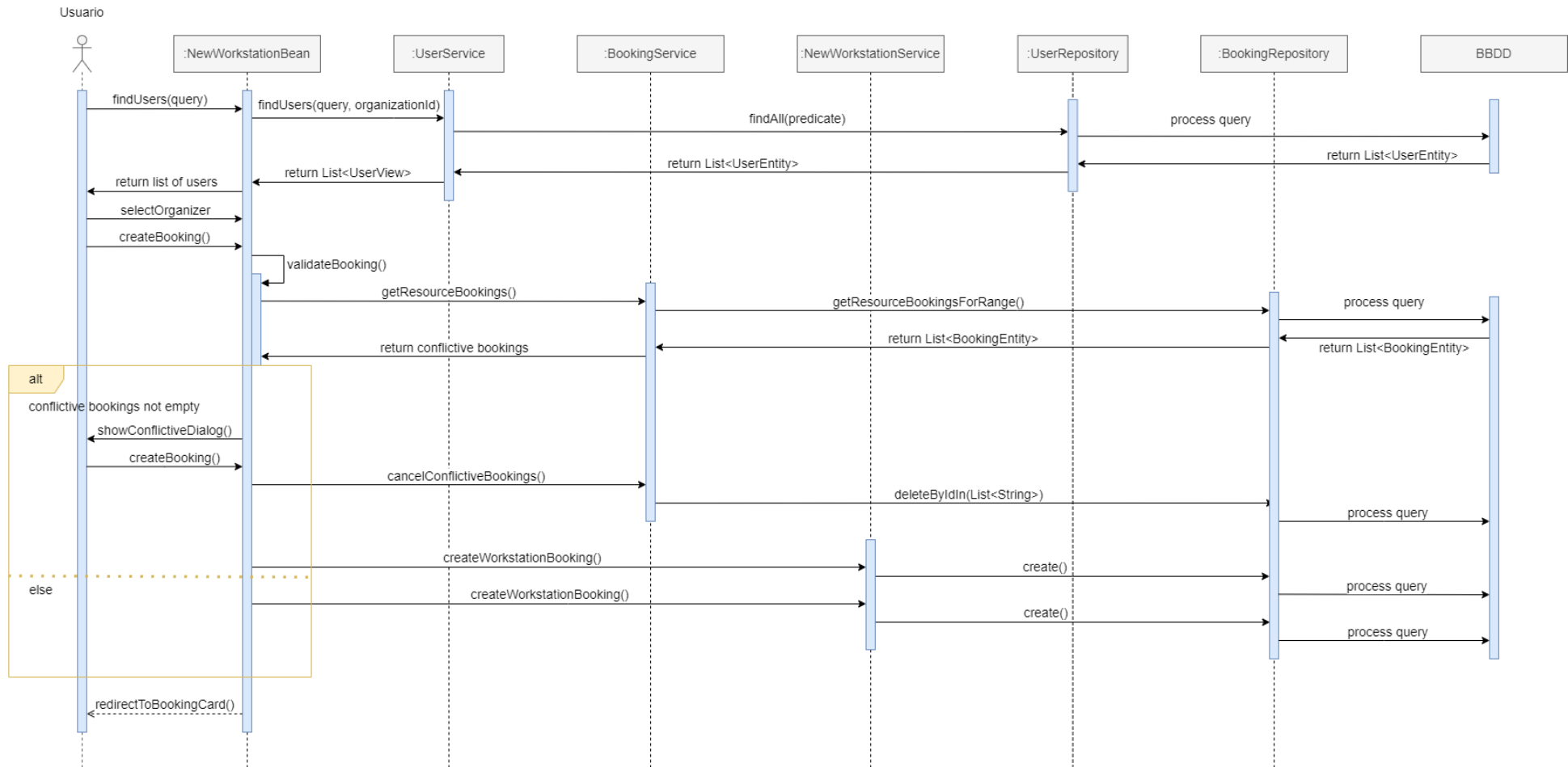
4.1.2. Crear reserva de puesto de trabajo

Tabla 2. Caso de uso “Crear reserva de puesto de trabajo”.

ID y nombre	UC-02. Crear reserva de puesto de trabajo
Descripción	El usuario que ha iniciado sesión debe ser capaz de crear una reserva de puesto de trabajo en el puesto de trabajo elegido.
Disparador	El usuario pulsa el botón “Nueva reserva” presente en el panel lateral mostrado al pulsar en un puesto de trabajo del listado de recursos.
Actor	Usuario con sesión iniciada.
Precondición	El usuario ha seleccionado un puesto de trabajo del listado de recursos.
Resultado	La reserva de puesto de trabajo ha sido creada.
Escenario principal	<ol style="list-style-type: none">1. El sistema redirige al usuario al formulario de crear reserva de puesto de trabajo.2. El usuario selecciona un organizador.3. El usuario selecciona una fecha y hora de inicio y fin válidas.4. El usuario pulsa el botón “Crear reserva”.5. El sistema valida los datos introducidos y crea la reserva en la base de datos.

	6. El sistema redirige al usuario a la ficha de reserva creada.
Escenario alternativo	<p>5a. El sistema detecta que el recurso a reservar ya está reservado para la fecha y hora seleccionada por el usuario.</p> <p>5a1. El sistema muestra un diálogo avisando al usuario que proceder con la creación conlleva anular las reservas existentes del recurso.</p>
Excepciones	<p>El usuario no ha seleccionado un organizador: Se impide la creación de la reserva y se le muestra un mensaje de error al usuario.</p> <p>La fecha y hora de inicio es anterior a la fecha y hora actual o bien la hora de fin de la reserva es anterior a la hora de inicio de la reserva: Se impide la creación de la reserva y se le muestra un mensaje de error al usuario.</p>

Figura 10. Diagrama de secuencia del caso de uso "Crear reserva de puesto de trabajo".



4.1.3. Crear reserva de sala de reuniones

Tabla 3. Caso de uso "Crear reserva de sala de reuniones".

ID y nombre	UC-03. Crear reserva de sala de reuniones
Descripción	El usuario que ha iniciado sesión debe ser capaz de crear una reserva de puesto de trabajo en el puesto de trabajo elegido.
Disparador	El usuario pulsa el botón “Nueva reserva” presente en el panel lateral mostrado al pulsar en un puesto de trabajo del listado de recursos.
Actor	Usuario con sesión iniciada.
Precondición	El usuario ha seleccionado una sala de reuniones del listado de recursos.
Resultado	La reserva de sala de reuniones ha sido creada.
Escenario principal	<ol style="list-style-type: none">1. El sistema redirige al usuario al formulario de crear reserva de sala de reuniones.2. El usuario selecciona un organizador.3. El usuario selecciona una fecha y hora de inicio y fin válidas.4. El usuario introduce un título de reserva.5. El usuario introduce una descripción de reserva.6. El usuario añade invitados a la reserva.

	<p>7. El usuario pulsa el botón “Crear reserva”.</p> <p>8. El sistema valida los datos introducidos y crea la reserva en la base de datos.</p> <p>9. El sistema redirige al usuario a la ficha de reserva creada.</p>
Escenario alternativo	<p>4a. El usuario no introduce un título de reserva.</p> <p>5a. El usuario no introduce una descripción de reserva.</p> <p>6a. El usuario no añade invitados a la reserva.</p> <p>5a. El sistema detecta que el recurso a reservar ya está reservado para la fecha y hora seleccionada por el usuario.</p> <p>5a1. El sistema muestra un diálogo avisando al usuario que proceder con la creación conlleva anular las reservas existentes del recurso.</p>
Excepciones	<p>El usuario no ha seleccionado un organizador: Se impide la creación de la reserva y se le muestra un mensaje de error al usuario.</p> <p>La fecha y hora de inicio es anterior a la fecha y hora actual o bien la hora de fin de la reserva es anterior a la hora de inicio de la reserva: Se impide la creación de la reserva y se le muestra un mensaje de error al usuario.</p>

4.2. Base de datos

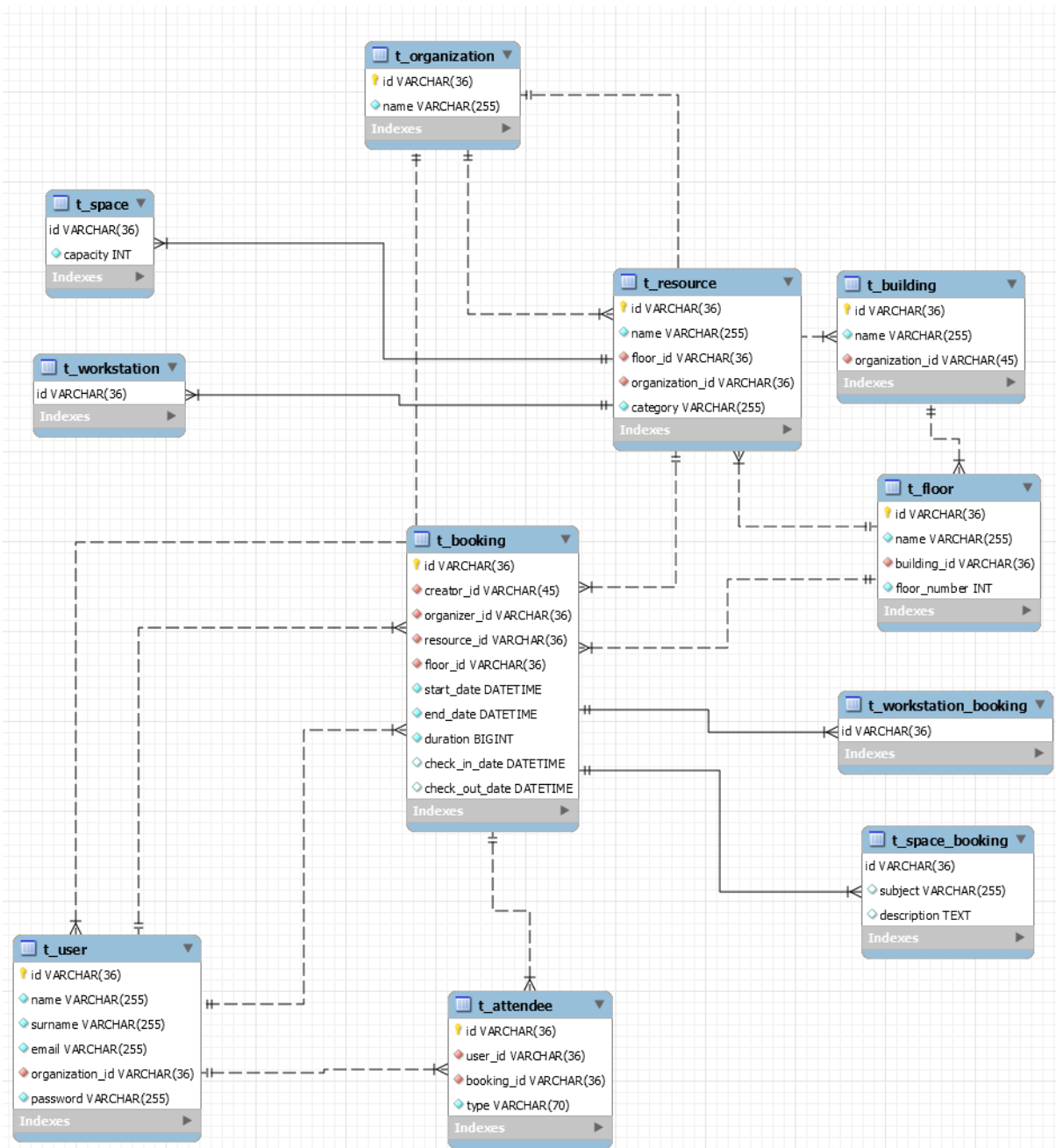
Tal y como se explica en el apartado 2.2.7, el software gestor de base de datos usado en el desarrollo de la aplicación es MySQL, y el software utilizado para crear la base de datos, las diferentes tablas y gestionar éstas es MySQL Workbench.

La base de datos creada para esta aplicación utiliza una codificación UTF-8 para los datos presentes en las diferentes tablas.

4.2.1. Diagrama entidad-relación

Para facilitar la visualización de la estructura de la base de datos, se ha generado un diagrama entidad-relación con la ayuda de MySQL Workbench.

Figura 11. Diagrama entidad-relación de la base de datos.



4.2.2. Especificación de las tablas

La base de datos contiene un total de 11 tablas que serán usadas en diferentes puntos de la aplicación y que se relacionan entre ellas de diferentes maneras. A continuación, se especifica en detalle cada una de ellas.

Es importante mencionar que todas las tablas presentes en la base de datos cuentan con un identificador único que sirve de clave primaria. Este identificador es un UUID (**U**niversally **I**dentifier, por sus siglas en inglés) de cuarta generación y de 36 caracteres de longitud generado automáticamente por la librería Spring JPA proporcionada por el framework de Spring. Este identificador sirve como clave primaria de la tabla. El tipo de dato del identificador en cada tabla es VARCHAR(36), significando esto que permite una cadena de longitud variable, y con un máximo de 36 caracteres.

Por simplicidad, no se especifica en cada definición de cada tabla que posee un identificador propio, ya que todas lo poseen, solo en casos en los que sea relevante por su relación con otras tablas.

4.2.2.1. t_organization

Tabla encargada de guardar los datos de las diferentes organizaciones que sirven como clientes de nuestra aplicación, y cuyos espacios de trabajo gestionamos.

Podemos considerar esta tabla como la tabla principal, ya que el resto de las tablas presentes en la base de datos están relacionadas de forma directa o indirecta con esta tabla.

Sus campos son los siguientes:

- **name:** El nombre de la organización y/o cliente en concreto. Nos sirve para identificar a cada cliente, ya que un identificador, aunque útil y necesario, no es legible por los usuarios. El tipo de dato utilizado para este campo es VARCHAR(255).

4.2.2.2. t_building

Tabla que almacena los datos de los diferentes edificios donde se encuentren las oficinas del cliente y que se relaciona directamente con la tabla t_organization.

Sus campos son los siguientes:

- **name:** El nombre del edificio definido por el cliente, de forma que los usuarios puedan identificarlo sin necesidad de utilizar los identificadores únicos de cada registro, ya que no son prácticos para un usuario. El tipo de dato utilizado para este campo es VARCHAR(255).
- **organization_id:** El id de organización, que se relaciona con la tabla t_organization mediante clave foránea. Este identificador no puede ser nulo, por lo que un edificio siempre tendrá que estar asociado a una organización. Su tipo de dato es VARCHAR(36).

4.2.2.3. t_floor

Tabla que almacena los datos de las diferentes plantas presentes en los edificios de la organización. Cada registro presente en esta tabla está relacionado directamente con el edificio al que pertenezca.

Sus campos son los siguientes:

- **name:** El nombre de la planta definido por el cliente, de forma que los usuarios puedan identificarla sin necesidad de utilizar los identificadores únicos de cada registro, ya que no son prácticos para un usuario. El tipo de dato utilizado para este campo es VARCHAR(255).
- **floor_number:** Número de la planta respecto al edificio, especialmente útil si existen varias oficinas dentro de un mismo edificio. Este campo utiliza el tipo de dato INT y tiene una restricción de unicidad a nivel de edificio, significando esto que dentro de un mismo edificio no pueden existir dos plantas con el mismo número de planta.
- **building_id:** El id del edificio al que pertenece la planta, que se relaciona directamente con la tabla t_building usando una clave foránea. Este identificador no puede ser nulo, por lo que una planta siempre tiene que estar asociada a una organización. El tipo de dato es VARCHAR(36).

4.2.2.4. t_resource

Tabla que guarda la información de cada espacio de trabajo que la empresa desee gestionar desde nuestra aplicación web, sobre estos espacios de trabajo se gestionan las reservas.

Sus campos son los siguientes:

- **name:** El nombre del espacio de trabajo definido por el cliente, de forma que los usuarios puedan identificarla sin necesidad de utilizar los identificadores únicos de cada registro, ya que no son prácticos para un usuario. El tipo de dato utilizado para este campo es VARCHAR(255).
- **floor_id:** Identificador de la planta a la que pertenece el recurso o espacio de trabajo, que se relaciona directamente con la tabla floor usando una clave foránea. Este identificador no puede ser nulo, por lo que un recurso siempre está asociado a una planta. El tipo de dato es VARCHAR(36).
- **organization_id:** Identificador de la organización a la que pertenece el recurso o espacio de trabajo. Aunque ya estemos relacionando de forma indirecta el recurso de forma con la organización (ya que al relacionarlo con la planta lo estamos indirectamente relacionando con el edificio y a su vez con la organización) esta relación nos sirve a la hora de hacer consultas para ahorrar una cantidad de tiempo notable debido a que evitamos utilizar otras tablas mediante JOINS en las consultas. Se utiliza una clave foránea y este campo no puede ser nulo. El tipo de dato es VARCHAR(36).
- **category:** Enumerado con dos valores posibles: SPACE y WORKSTATION, que nos ayudan a la hora de realizar consultas desde Java y así evitar el uso de JOINS si queremos recuperar los recursos de un tipo u otro.

4.2.2.5. t_workstation

Tabla que sirve para definir espacios de trabajo de tipo “puesto”. Como característica especial respecto a las diferentes tablas se destaca el identificador de cada registro es el mismo que un identificador encontrado en la tabla t_resource, creando de esta forma un tipo de “herencia” entre tablas.

El motivo principal de la existencia de esta tabla en vez de usar un campo dentro de la propia tabla t_resource que diferencie los diferentes tipos de recursos o espacios de trabajo es que

si en un futuro se quieren añadir características específicas a los diferentes tipos bastaría con añadir campos nuevos a esta tabla, en vez de tener que realizar cambios notables en la arquitectura de base de datos, con lo que esto conlleva (problemas de compatibilidad con aplicaciones y/o servicios ya existentes dentro nuestro ecosistema).

El único campo de esta tabla es:

- **id:** Identificador de la tabla y clave primaria, con tipo de dato VARCHAR(36). Este campo se relaciona mediante clave foránea con el id de la tabla t_resource.

4.2.2.6. t_space

Tabla que sirve para definir espacios de trabajo de tipo “sala de reuniones”. Sigue la misma dinámica que la tabla t_workstation respecto al identificador y el motivo de su existencia en la base de datos es también el mismo.

El único campo de esta tabla es:

- **id:** Identificador de la tabla y clave primaria, con tipo de dato VARCHAR(36). Este campo se relaciona mediante clave foránea con el id de la tabla t_resource.
- **capacity:** Capacidad máxima de personas que puede acceder a la sala de reuniones. El tipo de campo es INT (entero) y no puede ser nulo.

4.2.2.7. t_user

En esta tabla se almacenan los diferentes trabajadores de una organización y que sirven como organizadores de una reserva, o bien como invitados si se trata de una reserva de una sala de reuniones.

Los campos de esta tabla son los siguientes:

- **name:** Nombre del usuario, con tipo de dato VARCHAR(255).
- **surname:** Nombre del usuario, con tipo de dato VARCHAR(255).
- **email:** Correo electrónico del usuario, con tipo de dato VARCHAR(255) y que es único por organización.

4.2.2.8. t_booking

Esta tabla se puede considerar como el grueso principal de nuestra aplicación ya que es la que guardará los datos de cada reserva realizada por los usuarios.

Sus campos son los siguientes:

- **creator_id**: Identificador del usuario que crea la reserva desde la aplicación. No tiene por qué ser el mismo usuario que organiza la reserva. El tipo de dato es VARCHAR(36) y no puede ser nulo.
- **organizer_id**: Identificador del usuario que servirá como organizador de la reserva. Este identificador se relaciona directamente con la tabla planta usando una clave foránea. El tipo de dato es VARCHAR(36) y no puede ser nulo.
- **resource_id**: El espacio de trabajo reservado, se relaciona con la tabla t_resource mediante clave foránea, utiliza el tipo de dato VARCHAR(36) y no puede ser nulo.
- **floor_id**: El id de la planta a la que pertenece el espacio de trabajo reservado, que se utiliza a la hora de realizar consultas para disminuir el tiempo que éstas tardan en completarse. Se relaciona con la tabla t_floor mediante clave foránea, utiliza el tipo de dato VARCHAR(36) y no puede ser nulo.
- **start_date**: Fecha y hora de inicio de la reserva, que se guarda en UTC+0 y con el formato por defecto de MySQL (YYYY-MM-DD HH:mm:ss). El tipo de dato es DATETIME (que como su propio nombre indica se utiliza para almacenar fechas y horas) y no puede ser nulo.
- **end_date**: Fecha y hora de fin de la reserva, también se guarda en UTC+0 como start_date y su tipo de dato es DATETIME.
- **duration**: Duración de la reserva en milisegundos, se calcula desde código a la hora de crear la reserva. El tipo de dato es BIGINT, que es utilizado para guardar datos números de gran tamaño y no puede ser nulo.
- **check_in_date**: Fecha y hora en la que se realizó el check-in (confirmación de asistencia). El tipo de dato es DATETIME y puede ser nulo, ya que un usuario puede no realizar el check in.
- **check_out_date**: Fecha y hora en la que se realizó el check out (el usuario confirma que abandona el espacio de trabajo). El tipo de dato es DATETIME y puede ser nulo, ya que un usuario puede no realizar el check out.

4.2.2.9. t_space_booking

Tabla específica para las reservas de salas de reuniones cuyo identificador coincide con un identificador de la tabla t_booking. De esta forma, e igual que se ha hecho con las tablas t_workstation y t_space, se consigue una forma de “herencia” entre tablas y permite que, si en un futuro se quieren añadir características a las reservas de tipos específicos con añadir campos nuevos sea suficiente, sin necesidad de añadir campos a la tabla principal de reservas (t_booking) que quizás otros tipos de reservas no aprovechan.

Los campos de esta tabla son los siguientes:

- **id:** Identificador de la tabla y clave primaria, con tipo de dato VARCHAR(36). Este campo se relaciona mediante clave foránea con el id de la tabla t_booking.
- **subject:** Título de la reserva. El tipo de dato utilizado es VARCHAR(255) y puede ser nulo.
- **description:** Descripción de la reserva. El tipo de dato utilizado es TEXT, que sirve para guardar una gran cantidad de texto. Este campo puede ser nulo.

4.2.2.10. t_workstation_booking

De igual forma que con la tabla t_space_booking, se crea esta tabla específica para las reservas sobre espacios de trabajo de tipo “puesto”.

Su único campo es:

- **id:** Identificador de la tabla y clave primaria, con tipo de dato VARCHAR(36). Este campo se relaciona mediante clave foránea con el id de la tabla t_booking.

4.2.2.11. t_attendee

Tabla encargada de almacenar los invitados de una reserva de sala de reuniones.

Sus campos son los siguientes:

- **id:** Identificador de la tabla y clave primaria, con tipo de dato VARCHAR(36).
- **user_id:** Identificador del usuario invitado a la sala de reuniones, se relaciona mediante clave foránea a la tabla t_user. El tipo de dato es VARCHAR(36). No puede ser nulo

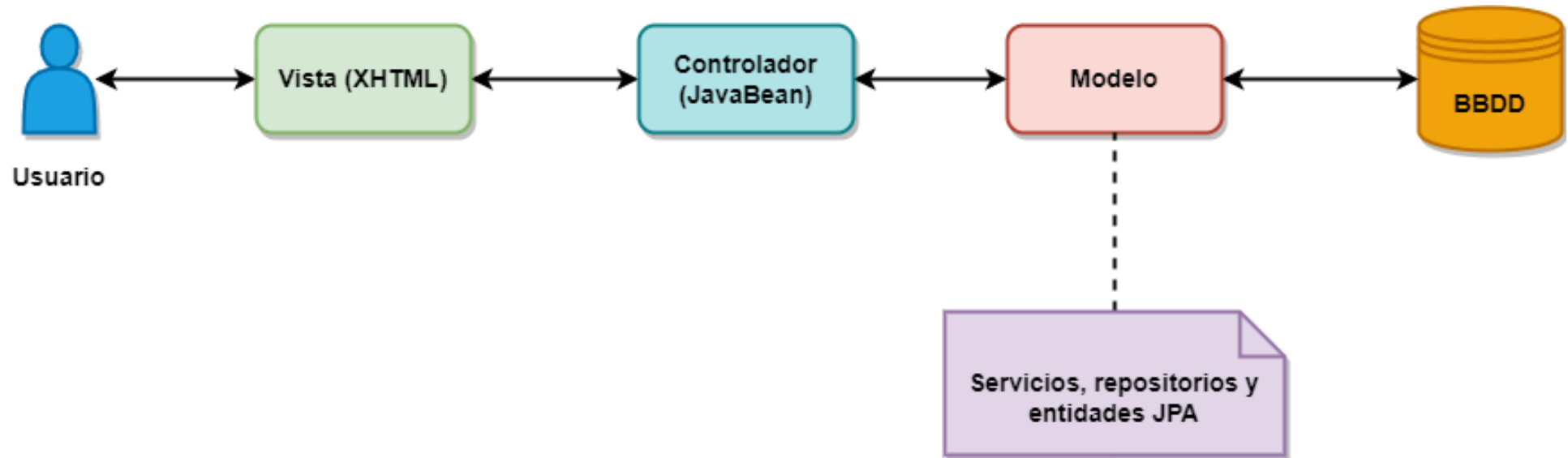
- **booking_id:** Identificador de la reserva a la que está ligado este invitado y que se relaciona mediante clave foránea a la tabla t_booking. El tipo de dato es VARCHAR(36). No puede ser nulo.
- **type:** Tipo de invitado, puede tener los valores OPTIONAL, REQUIRED . El primer valor indica que la asistencia del invitado es opcional y el segundo que la asistencia es requerida. El tipo de dato es VARCHAR(70) y no puede ser nulo.

4.3. Front-end y back-end

En este apartado se explica en detalle cada vista (front-end) de la aplicación y cómo se conecta con el back-end. Para cada vista se incluye una imagen de dicha vista.

Es importante mencionar que la aplicación está diseñada usando el patrón de diseño MVC (Modelo – Vista – Controlador). Esto significa que para cada vista (archivo xhtml) existe un controlador (un JavaBean con el que se conectará la vista) y un modelo en forma de clases servicio (ya que tienen la anotación `@Service` de Spring) que son las que se conectan con los repositorios JPA y con las entidades que representan las tablas de la base de datos. El controlador se comunica con el modelo, este realiza las acciones oportunas sobre la base de datos y devuelve los resultados de dichas acciones al controlador, que a su vez notifica a la vista y ésta al usuario.

Figura 12. Diagrama de arquitectura de la aplicación.



4.3.1. Diseño de las vistas de la aplicación

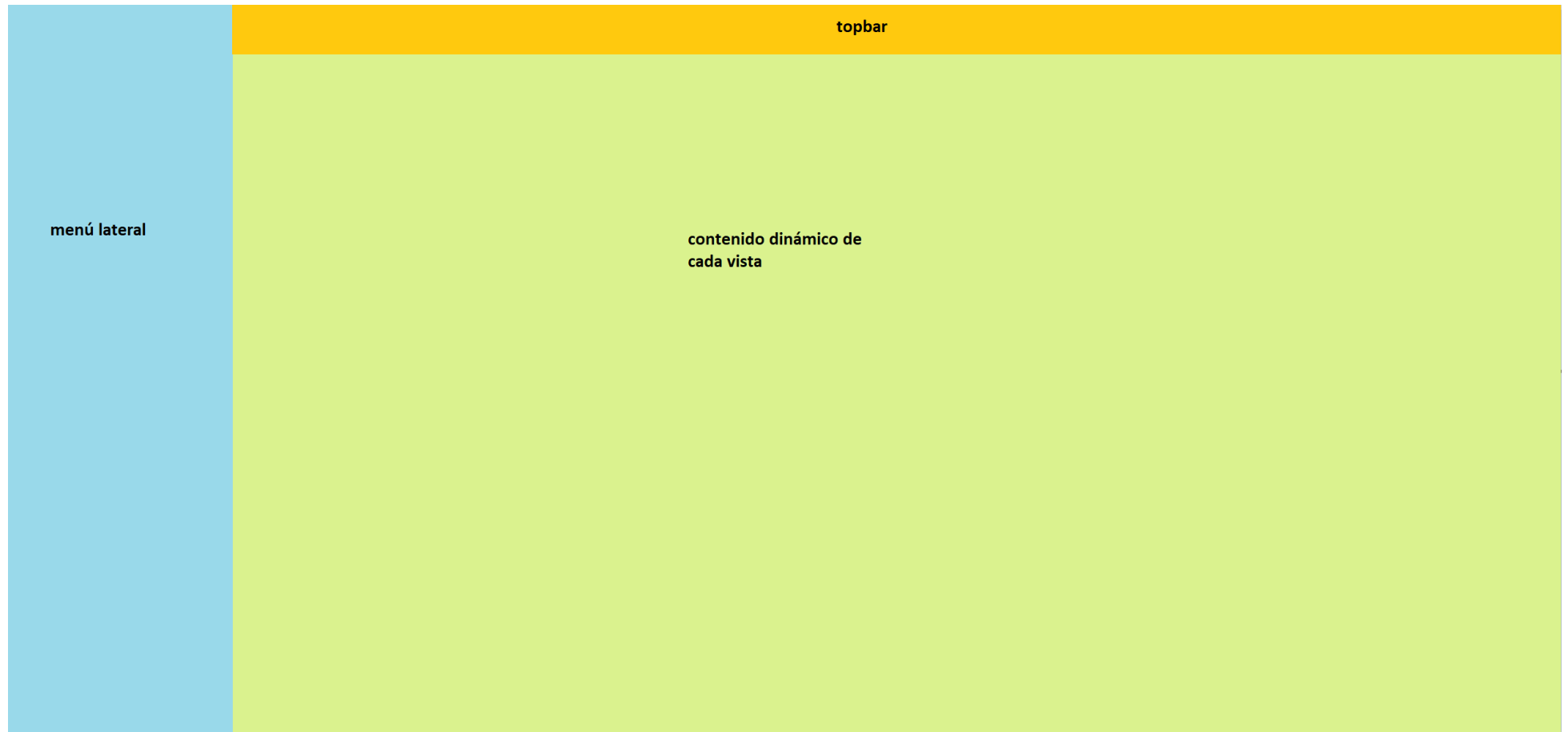
Todas las vistas presentes, excluyendo las vistas relacionadas con el login y logout, utilizan una plantilla como base para evitar tener que incluir o definir partes estáticas (que no van a cambiar independientemente de en qué vista se encuentre el usuario) en cada vista que creemos.

Esta plantilla ha sido creada específicamente para esta aplicación y define un menú lateral, una barra superior o topbar y un espacio para el contenido en el que se muestra cada vista presente en la aplicación. Es en esta plantilla donde también se incluyen los estilos CSS globales de la aplicación.

Los logos utilizados a lo largo de la aplicación han sido proporcionados por la página web Kadence WP (Kadence WP, s. f.), y que poseen una licencia CC0, por lo que su uso es gratuito. (Creative Commons, s. f.)

Las imágenes utilizadas en la aplicación han sido proporcionadas por la página web Unsplash, tanto la utilizada para los puestos de trabajo (Banks, 2020) como para las salas de reuniones (Deaner, 2018). La licencia de Unsplash permite usar las imágenes de su aplicación de forma gratuita. (Unsplash, s. f.)

Figura 13. Estructura de la interfaz en base a la plantilla creada para la aplicación.



4.3.2. Login (Inicio de sesión) y Logout

El inicio sesión es la primera vista que se encuentra un usuario final que quiera usar la aplicación, siendo imposible acceder al resto de vistas mientras el usuario no esté autenticado.

El usuario debe introducir su email y contraseña registrada en la base de datos. Una vez introducido, y si los datos son correctos, el usuario es autenticado y redirigido a la vista inicial de la aplicación. En caso de que el usuario introduzca incorrectamente sus datos, se le indica tal cosa y puede reintentar el inicio de sesión. El usuario puede realizar el número de intentos de inicio de sesión que considere.

Una vez autenticado, el usuario puede realizar el logout en cualquier momento que desee, siendo redirigido a una pantalla que le indicará que el logout se ha realizado correctamente, y por tanto siendo necesario realizar el proceso de inicio de sesión de nuevo si se quiere volver a acceder a la aplicación.

Es importante mencionar que en esta aplicación no se permite que el usuario se registre, ya que en el ecosistema de Bookker el alta de la organización, así como sus recursos y usuarios se realizan desde otra aplicación web independiente. De esta forma, únicamente los usuarios ya registrados en el sistema podrán iniciar sesión.

Para la gestión del inicio de sesión se ha utilizado la librería Spring Security, que proporciona una serie de herramientas que son capaces de reconocer los componentes de email y contraseña de la vista (siempre que se les establezcan los identificadores necesarios a los componentes que Spring necesita para poder reconocerlos) y realizar las consultas apropiadas en base de datos para validar que los datos introducidos son correctos.

En el proyecto existe una clase que implementa la interfaz `UserDetailsService`. Esta clase sobrescribe el método `loadUserByUsername`, que acepta un `String` como parámetro y que es el email en nuestro caso (pero también podría ser el nombre de usuario) introducido por el usuario. Mediante el repositorio JPA dedicado a la entidad `UserEntity` (y por tanto la tabla `t_user` de la base de datos) creado en la capa de persistencia, se busca al usuario en la base de datos. Si se encuentra al usuario, se crea un objeto que sirve como representación de usuario y que se almacena en el contexto de seguridad de Spring.

Para poder acceder a los datos de este usuario (incluidos los datos de la organización a la que pertenece) a lo largo del ciclo de vida de la aplicación se ha creado un objeto llamado

MyUserPrincipal que implementa la clase UserDetails y que sirve para luego poder castear el objeto devuelto por el contexto de autenticación a nuestra clase MyUserPrincipal.

La instancia de este objeto MyUserPrincipal es almacenada en un Bean (controlador) cuyo *scope* (ámbito o alcance) es de sesión, por lo cual siempre está vivo y con datos durante todo el ciclo de vida de la aplicación. El nombre de este Bean es SessionBean y nos sirve para recoger los datos del usuario que ha iniciado sesión, así como los datos de la organización a la que pertenece.

Spring Security proporciona una página de login, pero para el desarrollo de aplicación se han creado vistas de login, error de login y logout personalizadas.

Para configurar Spring Security es necesario crear una clase Java que extienda de la clase **WebSecurityConfigurerAdapter**, que nos obliga a sobrescribir los métodos *configure* y *passwordEncoder*. La clase creada que extiende de la clase proporcionada por Spring ha sido llamada **CustomSecurityConfig**.

El método *configure* nos permite configurar a que URLs tendrán acceso los usuarios, que página se usa de login y de logout, así como que página sirve como error de login para cuando un usuario introduzca unos datos inválidos al intentar iniciar sesión.

En el caso de nuestra aplicación, cualquier usuario puede acceder a los siguientes directorios, aunque no esté autenticado:

- Directorio raíz "/", ya que si no permitimos el acceso a éste ningún usuario podrá acceder al login.
- Vista de login y todas su subvistas ("/login*"), para que el usuario pueda acceder a la vista de login. El asterisco sirve para indicar que también sea posible acceder a cualquier vista independientemente de lo que aparezca detrás de "login" en la URL. Esto es útil ya que, por ejemplo, la vista de error de login utiliza parámetros en la url para indicar que ha ocurrido un error en el login, de tal forma que la URL para este caso de uso es "/login.xhtml?error=true". Si no ponemos el asterisco, sería necesario definir cada subvista, lo cual es un trabajo tedioso y que, si se añade o cambia algo, nos forzaría a cambiarlo también en la configuración.
- images, que es donde se guardan las diferentes imágenes utilizadas en la aplicación, y entre ellas imágenes utilizadas en el inicio de sesión.

- `css`, donde se guardan todas las hojas de estilos que se usan a lo largo de la aplicación.
- `javax.faces.resource`, para acceder a la librería de PrimeFaces y sus componentes usados en la vista.

Por otra parte, denegamos acceso directo a cualquier ruta que contenga un guion bajo, ya que en el proyecto se han creado archivos XHTML que comienzan con un guion bajo ya que forman parte de una vista más grande, y que se abstraen a otros archivos para conseguir una mejor legibilidad del código XHTML y que posteriormente se incluyen en dicha vista más grande.

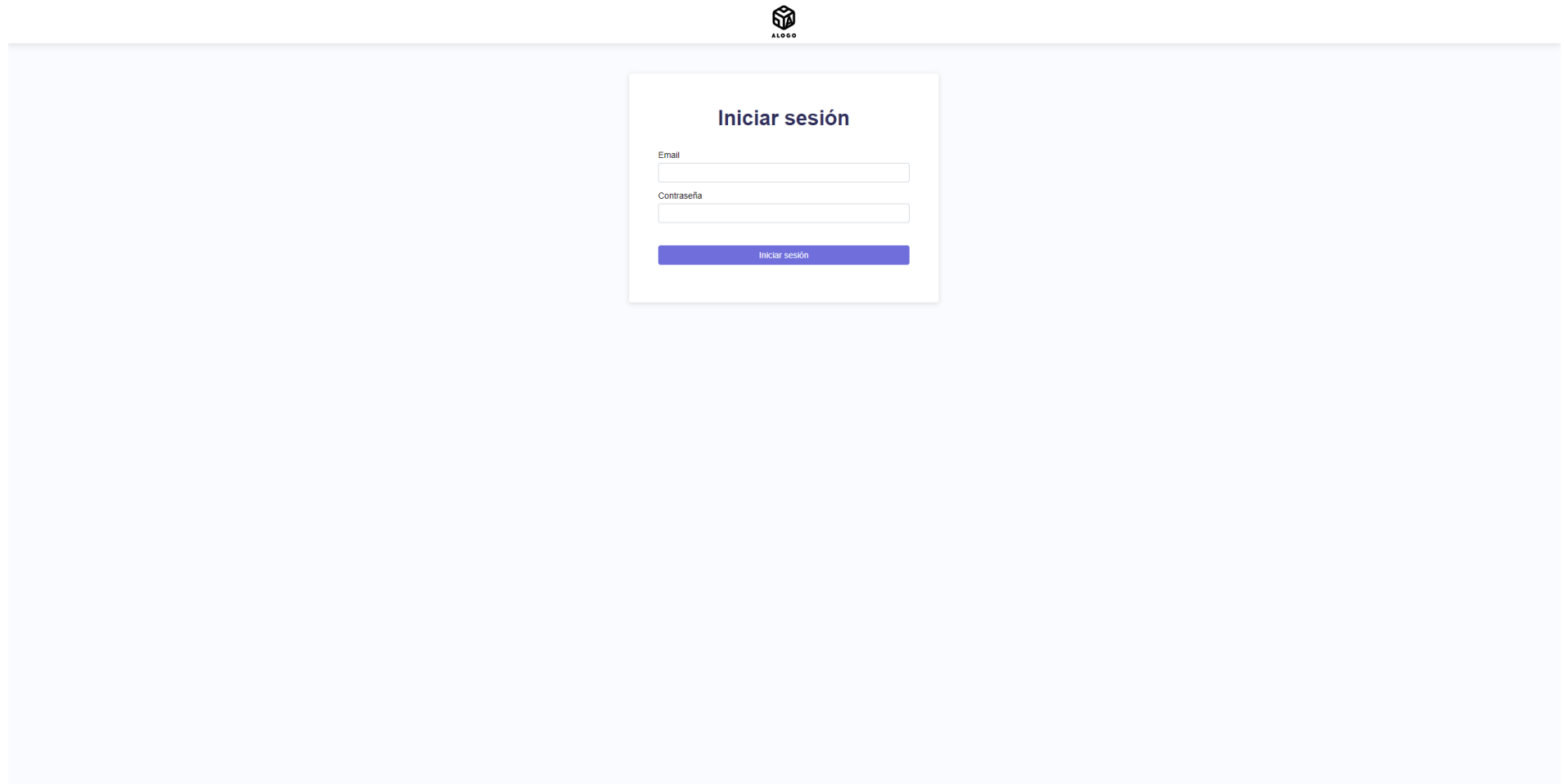
Para el resto de las vistas y/o directorios, el usuario debe estar autenticado para poder acceder.

Desde este método *configure* también le indicamos cual queremos que sea nuestra página de login, de logout y de error de login. Además, también borramos las cookies con nombre `JSESSIONID` e invalidamos la sesión del usuario cuando éste realice el logout.

Por último, indicamos que el máximo de sesiones por navegador es de una sesión.

Mediante el método *passwordEncoder* indicamos a Spring Security el codificador de contraseñas a utilizar y que sirve para descodificar la contraseña presente en la base de datos. Para esta aplicación se utiliza **BCryptPasswordEncoder**.

Figura 14. Vista de inicio de sesión.



The image shows a login form titled "Iniciar sesión" (Log in) centered on a light blue background. At the top center of the page is a small logo consisting of a cube with the word "ALOGO" underneath it. The login form itself is a white rectangle with a thin border. It contains the title "Iniciar sesión" in bold dark blue text. Below the title are two input fields: the first is labeled "Email" and the second is labeled "Contraseña" (Password). Both labels are in a small, light gray font. At the bottom of the form is a solid blue button with the text "Iniciar sesión" in white.

Figura 15. Vista de error de inicio de sesión.

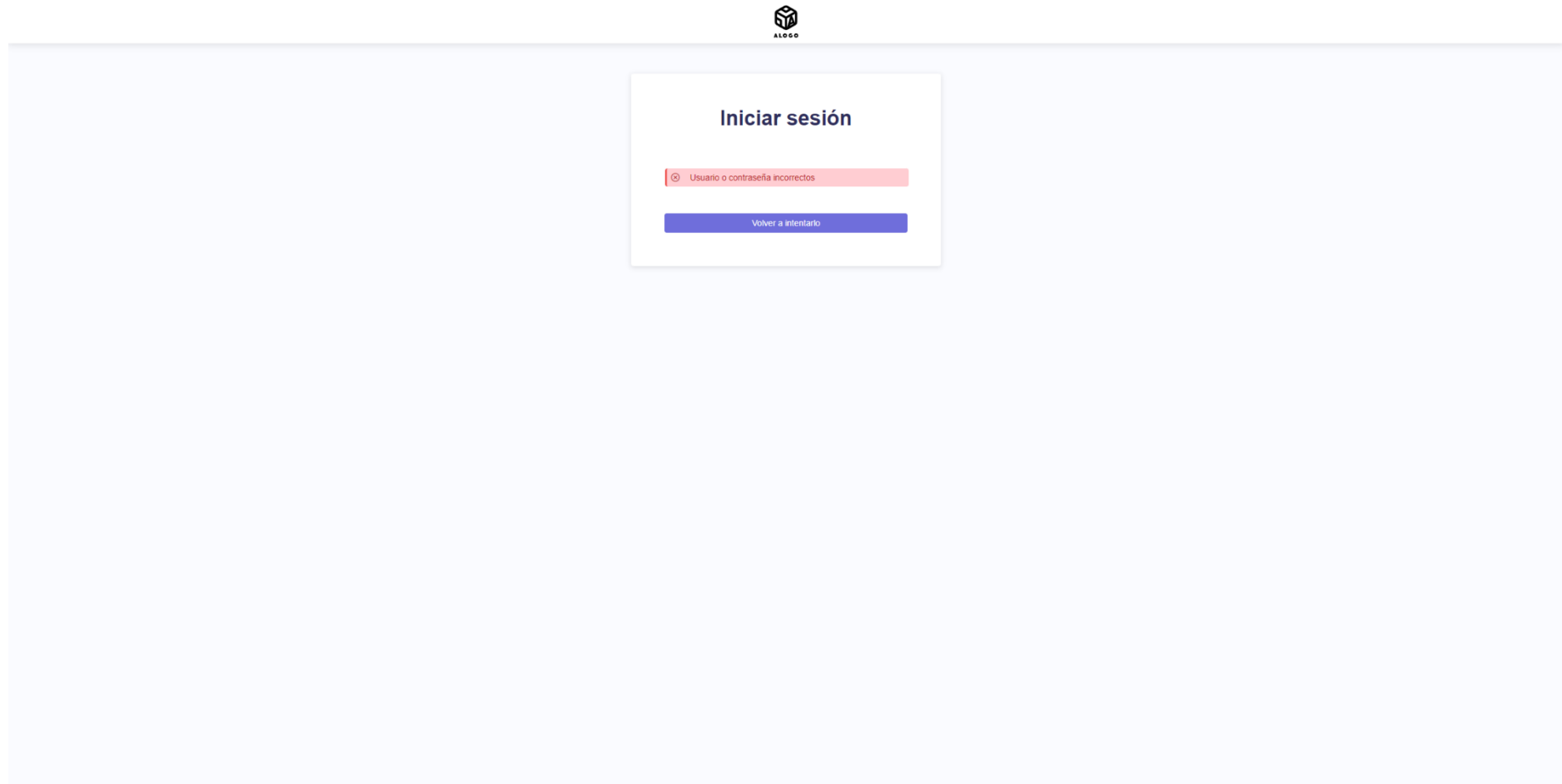
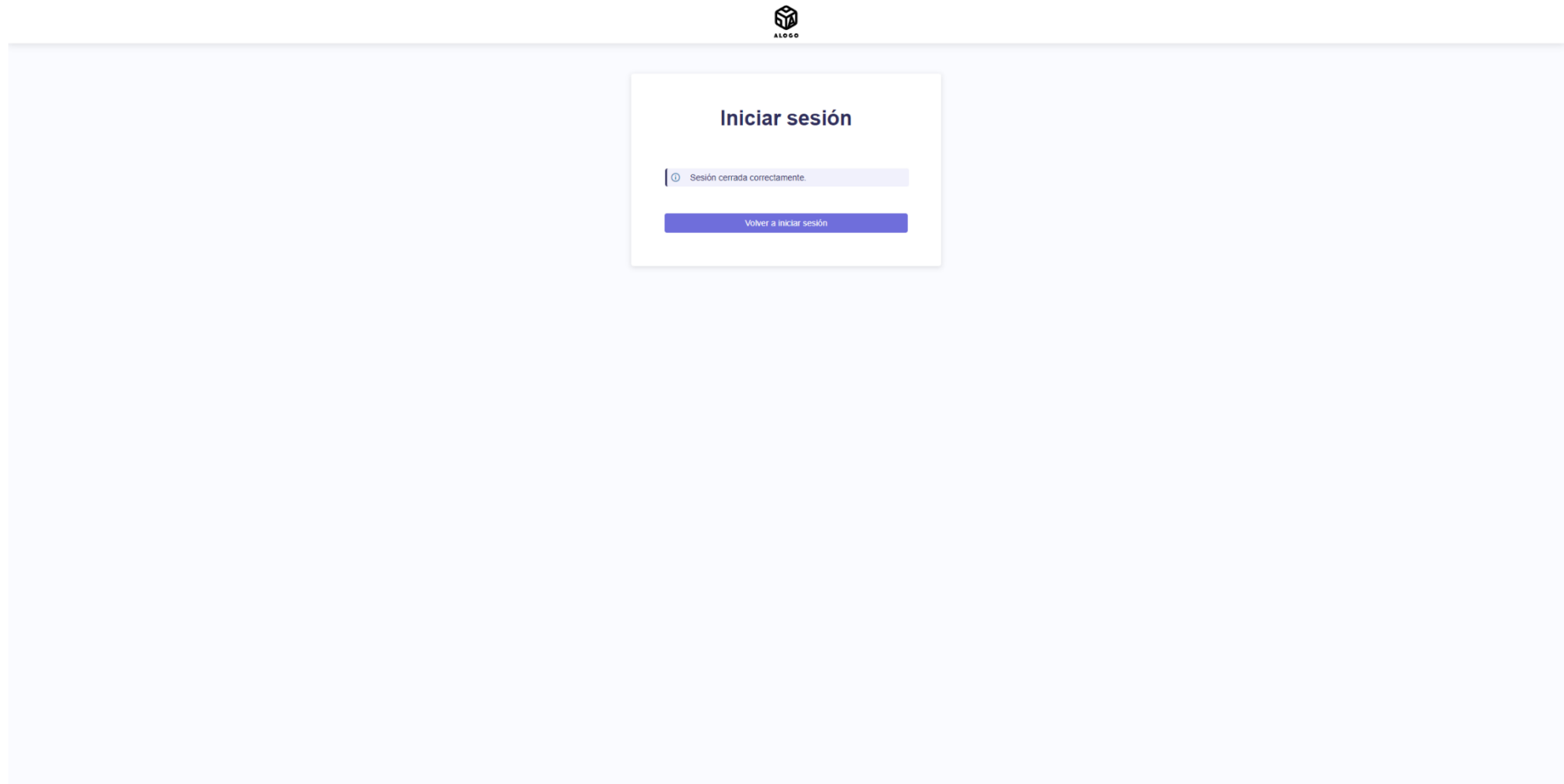


Figura 16. Vista de logout.



4.3.3. Búsqueda

Esta es la vista a la que se redirigen los usuarios que hayan iniciado sesión correctamente, y es la primera vista en la que ya podemos visualizar la estructura de la plantilla comentada en apartados anteriores.

En esta vista el usuario puede seleccionar un edificio de los disponibles, una planta de las que posee el edificio seleccionado y una fecha en la que desee buscar la ocupación de los recursos de la planta seleccionada.

El selector de planta está desactivado hasta que el usuario seleccione un edificio, y su contenido es actualizado dinámicamente cuando el usuario cambie de edificio seleccionado.

El selector de fecha se auto rellena con la fecha actual una vez el usuario seleccione una planta de las disponibles, permitiendo cambiar la fecha a discreción del usuario, teniendo como único límite que la fecha mínima sea el día actual.

El botón de “Buscar” se habilita una vez el usuario seleccione una planta, permitiendo avanzar a la siguiente vista.

Si el usuario deselecciona la planta o el edificio en cualquier momento, tanto el selector de fecha como el botón de búsqueda son deshabilitados de nuevo.

En el menú lateral podemos ver uno de los logos utilizados para la aplicación, así como un menú donde se muestra el nombre de la organización a la que pertenece el usuario que ha iniciado sesión y una opción dentro del menú llamada “Reservas”, que al ser pulsada redirige a esta vista principal.

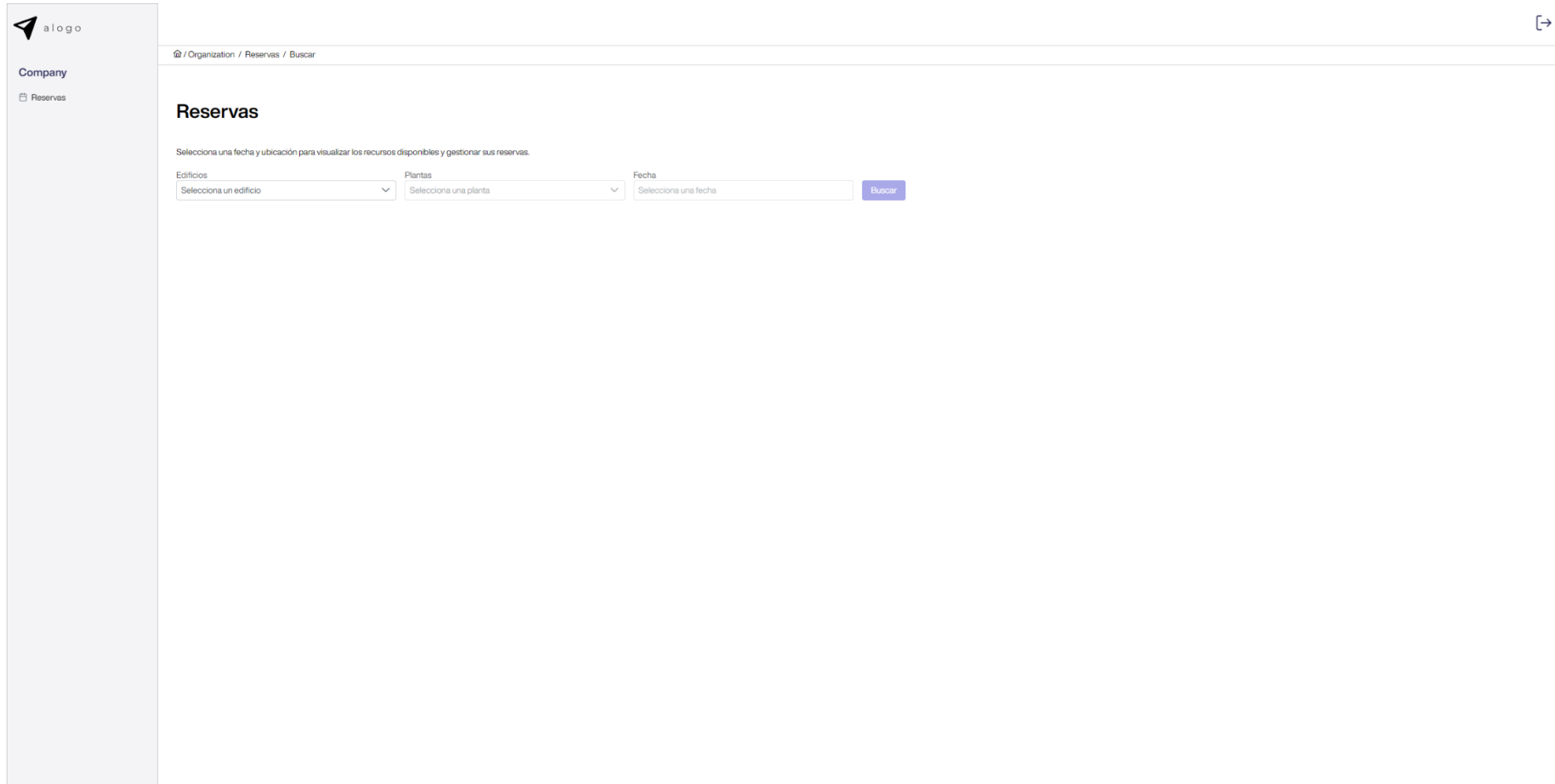
En el topbar podemos ver un icono en la parte derecha desde el que el usuario puede hacer log out.

El XHTML asociado a esta vista es **search.xhtml**, y se conecta al controlador (Bean) llamado **SearchBean**. Este controlador es el que conecta con los diferentes modelos (Llamados services por la anotación **@Service** de Spring). En el caso de **SearchBean**, tiene una instancia del bean de sesión llamado **SessionBean** para obtener el identificador de la organización a la que pertenece este usuario, y de esta forma obtener los edificios que pertenecen a dicha organización. Para recuperar los edificios y las plantas **SearchBean** utiliza el modelo o servicio llamado **SearchService**, que a su vez tiene instancias de los repositorios necesarios para

recuperar los edificios y las plantas de la base de datos (**BuildingRepository** y **FloorRepository**).

Estos repositorios JPA trabajan con las entidades de persistencia, pero para añadir una capa de abstracción extra, se han creado objetos vista para evitar que los controladores trabajen directamente con dichos objetos de persistencia. Estos objetos vista serán los utilizados en los diferentes controladores y vistas XHTML, y contienen los campos necesarios para el funcionamiento correcto de la vista.

Figura 17. Vista de búsqueda.



The screenshot displays the search interface of a workspace reservation application. On the left is a vertical sidebar with the 'a l o g o' logo and a 'Company' section containing a 'Reservas' link. The main content area features a breadcrumb trail 'Organization / Reservas / Buscar' and a title 'Reservas'. Below the title is a descriptive text: 'Selecciona una fecha y ubicación para visualizar los recursos disponibles y gestionar sus reservas.' The search form consists of three dropdown menus: 'Edificios' (with 'Selecciona un edificio' selected), 'Plantas' (with 'Selecciona una planta' selected), and 'Fecha' (with 'Selecciona una fecha' selected). A blue 'Buscar' button is positioned to the right of these fields. A share icon is located in the top right corner of the page.

4.3.4. Listado de recursos

Una vez el usuario pulsa en el botón buscar en la vista de “Búsqueda”, se le redirige a esta vista del listado de recursos.

Como se ha diseñado la aplicación de tal forma que cada vista es independientemente de las demás, y cada una usa su controlador propio, necesitamos comunicar entre esta vista y la de búsqueda el edificio, planta y fecha que ha seleccionado el usuario en la vista anterior. Para conseguir esto, se han utilizado parámetros en la URL, de la misma forma que se añadirían parámetros en una petición a una API REST.

El XHTML asociado a esta vista es **resources.xhtml**, su controlador (Bean) es **ResourceBean** y los servicios utilizados por el bean son **ResourceListService** (para recuperar los recursos de una planta, las reservas de cada recurso, calcular su disponibilidad y calcular la leyenda de ocupación situada encima de la tabla), y **BookingService** (utilizada para realizar acciones sobre las reservas individuales de cada recurso).

Los parámetros presentes en la URL son los siguientes:

- **date:** Fecha seleccionada por el usuario.
- **buildingId:** Identificador del edificio seleccionado por el usuario.
- **floorId:** Identificador de la planta seleccionada por el usuario.

Por lo tanto, un ejemplo de URL podría ser:

- `/resources.xhtml?date=20230703&buildingId=de1cc311-0804-4be6-8a07-f0a224357c3b&floorId=0e81a468-401b-494c-a338-aa3b508c135f`

Estos parámetros, al estar presentes en la URL, pueden ser modificados por el usuario y actualizar la página para reflejar esos cambios. En caso de que alguno de los tres parámetros no sea encontrado por el Bean asociado a esta vista se redirige a la vista inicial de búsqueda. Esto también ocurre en caso de que los valores introducidos manualmente sean erróneos (por ejemplo, que el valor introducido como identificador del edificio no se corresponda con ningún edificio de la organización del usuario que ha iniciado sesión, o que directamente no exista en la base de datos).

El objetivo de esta vista es mostrar en un listado los recursos presentes en la planta seleccionada por el usuario, así como su ocupación para el rango de horas introducido (las

reservas que pueda tener cada recurso). El rango de horas utilizado por defecto es la hora actual redondeada al cuarto de hora más cercano (15, 30, 45 o en punto), ya que solo permitimos la búsqueda en tramos de 15 minutos, y hasta el final del día (23:45). Este rango de horas puede ser modificado desde el panel lateral dedicado a los filtros.

Se sigue permitiendo seleccionar un edificio, planta y fecha como en la vista de búsqueda, pero además podemos filtrar los recursos del listado introduciendo el nombre del recurso que queramos buscar en concreto, mostrando la aplicación los recursos cuyo nombre contenga el valor introducido por el usuario.

La leyenda de ocupación nos muestra mediante cuadrados de diferentes colores los siguientes valores posibles, así como el porcentaje de recursos que coinciden con el valor de la leyenda respecto al total de recursos del listado:

- **Libre:** El recurso no tiene reservas para el rango de fecha y horas seleccionado.
- **Parcialmente ocupado:** El recurso tiene reservas, pero la duración total de todas ellas no supera el rango de fecha y horas seleccionado, significando esto que el recurso dispone de algún tramo en el que puede ser reservado sin tener que cancelar una reserva existente.
- **Totalmente ocupado:** El recurso tiene reservas y la duración total de todas ellas es igual o superior al rango de fecha y horas seleccionado, significando esto que el recurso no dispone de algún tramo en el que pueda ser reservado sin tener que cancelar una reserva existente.

El listado de recursos es una tabla que permite ordenar los recursos por su nombre y por capacidad, facilitando la visualización de los datos. Esta tabla también muestra el edificio y la planta a la que pertenece el recurso (que coincidirá con el edificio y planta seleccionados en los selectores), así como el tipo de recurso (categoría), que puede ser “Puesto de trabajo” o “Sala de reuniones”. El número de elementos por página de la tabla se ha limitado a 15 para evitar un desbordamiento de ésta debido a un gran número de elementos. En caso de que una planta posea más de 15 recursos, la tabla se divide en páginas que podrán ser navegables desde la parte inferior de la tabla.

4.3.4.1. Mostrar filtros

Desde el botón “Mostrar filtros” los usuarios pueden acceder al panel lateral donde es posible modificar ciertos parámetros que harán que los datos de la tabla cambien. Este panel lateral permite al usuario modificar el rango de horas en el que se buscan las reservas de los recursos, así como filtrar los propios recursos de la planta mediante su categoría (mostrando todos los recursos que es el valor por defecto, sólo los puestos de trabajo o sólo las salas de reuniones). También es posible filtrar por la capacidad de un recurso, mostrando solo los recursos cuya capacidad sea igual o mayor al valor seleccionado.

Mediante el botón “Mostrar resultados”, el usuario puede aplicar el rango de horas y los filtros seleccionados, actualizando los porcentajes de ocupación (si procede) y los recursos presentes en la tabla.

Con el botón “Quitar filtros” el usuario puede desactivar el filtro de categoría y capacidad, conservando el rango de horas introducido. Esta acción, de igual forma que “mostrar resultados”, actualiza los porcentajes de ocupación (si procede) y los recursos presentes en la tabla.

El archivo XHTML asociado a esta vista es **_resourceFilters.xhtml**, y como se puede observar, este archivo contiene un guion bajo al inicio de su nombre. Esto es para evitar que los usuarios puedan acceder directamente al archivo desde la URL del navegador, ya que este archivo existe como añadido sobre la vista principal de listado de recursos. El bloqueo de acceso se hace mediante la configuración de Spring Security tal y como se comentó anteriormente.

Al ser una parte incluida en la vista principal **resources.xhtml**, el bean utilizado por la vista de filtros es también **ResourceBean**.

4.3.4.2. Panel lateral de un recurso específico

Cuando el usuario pulsa en una fila de la tabla, se despliega un panel lateral con la información del recurso seleccionado, una imagen representando el tipo de recurso seleccionado (puesto de trabajo o sala de reuniones) así como una tabla con la información de las reservas presentes para la fecha y rango de horas seleccionadas.

En esta tabla se muestra el horario de la reserva, así como el usuario organizador de ésta. Para cada reserva el usuario podrá realizar varias acciones:


- **Check in:** Realiza el check in sobre la reserva con la hora actual. Solo se puede realizar una vez por reserva, siendo la opción desactivada una vez se haya realizado el check in.
- **Check out:** Realiza el check out sobre la reserva con la hora actual. Solo se puede realizar una vez por reserva y si se ha realizado el check in previamente. Una vez realizado el check out, se desactiva la opción.
- **Editar reserva:** Se redirige a una vista específica donde se pueden editar diferentes parámetros de la reserva.
- **Anular reserva:** Se anula la reserva, eliminándola de la base de datos.

Desde el botón “Nueva reserva” el usuario puede acceder al formulario reserva en una vista nueva donde se podrán introducir los datos necesarios para completar el proceso de creación de reserva. El botón “Atrás” cierra el panel lateral.

El archivo XHTML asociado a esta vista es **_resourceSidebar.xhtml**, y de igual forma que con los filtros, posee un guion bajo al comienzo del nombre del archivo para evitar el acceso directo por parte de los usuarios.

El bean con el que se comunica esta vista es **ResourceBean**.

Figura 18. Vista de listado de recursos.



Company

Reservas

Organization / Reservas / Listado

Reservas por recurso

Selecciona una fecha y ubicación para visualizar los recursos disponibles y gestionar sus reservas.

Edificio

Edificio A

Planta

Planta 1

Fecha

03-07-2023

Buscar recurso

🔍
















Mostrar filtros

Ocupación desde 03-07-2023 - 13:15 hasta 03-07-2023 - 23:45

Libre 100%

Totalmente ocupado 0%

Parcialmente ocupado 0%

Nombre	Capacidad	Edificio	Planta	Categoría
 PUESTO 21	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 SALA 4	20 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 10	8 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 17	30 persona(s)	Edificio A	Planta 1	Sala de reuniones
 PUESTO 9	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 SALA 12	12 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 7	10 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 6	4 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 19	15 persona(s)	Edificio A	Planta 1	Sala de reuniones
 SALA 14	20 persona(s)	Edificio A	Planta 1	Sala de reuniones
 PUESTO 23	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 PUESTO 2	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 PUESTO 25	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 PUESTO 28	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
 PUESTO 1	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo

<<

<

1

>

>>

Figura 19. Panel lateral mostrar filtros.

The screenshot displays a web application for managing workspace reservations. The main panel is titled "Reservas por recurso" and includes a sidebar with "Company" and "Reservas" links. The main content area shows a search bar and a table of available rooms. A side panel on the right, titled "Filtros", contains additional filters for time, category, and capacity.

Reservas por recurso

Selección una fecha y ubicación para visualizar los recursos disponibles y gestionar sus reservas.

Edificio: Edificio A | Planta: Planta 1 | Fecha: 03-07-2023

Buscar recurso

Ocupación desde 03-07-2023 - 13:15 hasta 03-07-2023 - 18:45

■ Libre 92% ■ Totalmente ocupado 0% ■ Parcialmente ocupado 7%

Nombre ↑↓	Capacidad ↑↓	Edificio	Planta
■ SALA 4	20 persona(s)	Edificio A	Planta 1
■ SALA 17	30 persona(s)	Edificio A	Planta 1
■ SALA 14	20 persona(s)	Edificio A	Planta 1
■ SALA 27	20 persona(s)	Edificio A	Planta 1

Filtros

Desde / Hasta: 13:15 - 18:45

Categoría: Sala de reuniones

Capacidad: 20+

Quitar filtros | Mostrar resultados

Figura 20. Panel lateral de un recurso específico.

The screenshot displays the 'Reservas por recurso' (Reservations by resource) interface. On the left, a sidebar contains the 'Company' logo and a 'Reservas' menu item. The main content area features a breadcrumb trail 'Organization / Reservas / Listado' and a title 'Reservas por recurso'. Below the title, a sub-header instructs users to 'Selecciona una fecha y ubicación para visualizar los recursos disponibles y gestionar sus reservas.' (Select a date and location to view available resources and manage their reservations). This is followed by filter inputs for 'Edificio' (Building) set to 'Edificio A', 'Planta' (Floor) set to 'Planta 1', and 'Fecha' (Date) set to '03-07-2023'. A search bar labeled 'Buscar recurso' is also present. Below the filters, a summary shows occupancy from '03-07-2023 - 13:15' to '03-07-2023 - 18:45': 'Libre 92%' (Free), 'Totalmente ocupado 0%' (Fully occupied), and 'Parcialmente ocupado 7%' (Partially occupied). A table lists resources with columns 'Nombre' (Name), 'Capacidad' (Capacity), 'Edificio' (Building), and 'Planta' (Floor). The table contains four entries: SALA 4 (20 capacity), SALA 17 (30 capacity), SALA 14 (20 capacity), and SALA 27 (20 capacity), all located in Edificio A, Planta 1. On the right, a detailed view for 'SALA 4' is shown, including a photo, name, location, and a table of reservations. The reservation table has columns 'Organizador' (Organizer) and 'Horario' (Schedule), with one entry for John Doe from 13:15 to 18:45. A context menu is open over this reservation, offering actions: 'Check in', 'Check out', 'Editar reserva' (Edit reservation), and 'Anular reserva' (Cancel reservation).

Nombre	Capacidad	Edificio	Planta
SALA 4	20 persona(s)	Edificio A	Planta 1
SALA 17	30 persona(s)	Edificio A	Planta 1
SALA 14	20 persona(s)	Edificio A	Planta 1
SALA 27	20 persona(s)	Edificio A	Planta 1

Organizador	Horario
John Doe	13:15 - 18:45

4.3.5. Creación de reserva de puesto de trabajo

Si el usuario pulsa en el botón de “Nueva reserva” desde el menú lateral de un puesto de trabajo, se le redirige a la vista de crear reserva de puesto de trabajo.

De la misma forma que para el resto de las vistas, es necesario comunicar mediante parámetros URL la fecha y hora de inicio, la fecha y hora de fin, así como el identificador del recurso seleccionado. Para lograr esto, se utilizan los siguientes parámetros:

- **start:** Fecha y hora de inicio de la reserva en milisegundos.
- **end:** Fecha y hora de fin de la reserva en milisegundos.
- **resourceId:** Identificador del recurso sobre el que se va a realizar la reserva.

Desde esta vista el usuario puede modificar la fecha y la hora de la reserva respetando los límites mínimos de fecha, siendo este límite la fecha actual. También es posible modificar la hora, no permitiendo completar la reserva si la fecha y hora de inicio es menor a la actual a la hora de completar el proceso de reserva.

Ya que la aplicación está orientada a crear reservas a terceros (el caso de uso principal sería el personal de secretaría de una compañía crea reservas al resto de empleados) es necesario seleccionar un organizador. Este organizador es un usuario registrado en el sistema y que pertenece a la misma organización que el usuario con sesión iniciada.

Para seleccionar un organizador, el usuario debe escribir en el campo de texto un mínimo de tres caracteres para que el sistema realice la consulta apropiada a la base de datos. Esta consulta busca usuarios cuyo nombre completo (nombre + apellidos) o email contiene la cadena de texto introducida por el usuario. Si el sistema encuentra resultados, devuelve un listado de usuarios que aparece debajo del componente de forma flotante, pudiendo el usuario seleccionar cualquiera de ellos.

Al pulsar el botón “Reservar”, el sistema realiza varias comprobaciones:

- Si el usuario no ha seleccionado un organizador para la reserva, se le muestra un mensaje de error impidiendo completar el proceso.
- Si la fecha u hora son inválidas (fecha y hora de inicio anterior a la actual u hora de fin anterior a hora de inicio) se le muestra un mensaje de error al usuario indicándolo.

- Se buscan las posibles reservas del recurso a reservar que coincidan total o parcialmente con la fecha y hora seleccionada por el usuario. En caso de que exista, se le muestra un diálogo al usuario indicando que continuar con el proceso anulará las reservas existentes (un recurso no puede tener dos reservas simultáneas que se solapen en el tiempo). Si el usuario decide proceder, se anulan las reservas existentes sobre ese recurso antes de crear la nueva reserva.

Si todas las comprobaciones previas a la creación de reservas han sido superadas correctamente, el sistema crea la reserva y el usuario es redirigido a la ficha de reserva de puesto de trabajo.

En caso de que el usuario desee cancelar el proceso de reserva, puede pulsar en el botón “Cancelar”, que lo redirige de vuelta a la vista del listado de recursos.

El archivo XHTML asociado a esta vista es **new_workstation.xhtml**, siendo **NewWorkstationBean** el bean con el que se conecta el XHTML. Este bean utiliza varios servicios:

- **NewWorkstationService**, para la creación de la reserva de puesto de trabajo.
- **UserService**, para la gestión de usuarios, como buscar usuarios en la base de datos cuando el usuario escribe caracteres en el campo de texto dedicado al organizador.
- **BookingService**, para la búsqueda de reservas conflictivas en las fechas y horas seleccionadas por el usuario.
- **ResourceService**, para recoger diferente información del recurso seleccionado, y así mostrar su información.

Figura 21. Creación de reserva de puesto de trabajo.

The screenshot shows a web application interface for creating a new job reservation. On the left is a sidebar with the 'a l o g o' logo and a 'Company' menu containing a 'Reservas' option. The main content area has a breadcrumb trail 'Organization / Reservas / Nueva reserva de puesto' and a title 'Reserva de PUESTO 21'. The form includes the following fields and elements:

- Location: 'Edificio A, Planta 1, PUESTO 21' with a location icon.
- Creator: 'Creada por: Aitor Gómez' with a user icon.
- Organizer: A label 'Organizador' above a text input field containing 'Buscar usuario'.
- Date: A label 'Fecha' above a date input field containing '4/7/2023'.
- Time: Two side-by-side time input fields. The first is labeled 'Hora de inicio' and contains '09:45'. The second is labeled 'Hora de fin' and contains '23:45'.
- Buttons: Two buttons at the bottom, 'Reservar' (highlighted in blue) and 'Cancelar' (disabled, in light gray).

A share icon is visible in the top right corner of the page.

Figura 22. Búsqueda de organizador en creación de reserva de puesto de trabajo.

The screenshot displays a web application interface for creating a new job reservation. On the left is a sidebar with a logo and navigation links for 'Company' and 'Reservas'. The main content area has a breadcrumb trail: 'Organization / Reservas / Nueva reserva de puesto'. Below this, the title 'Reserva de PUESTO 21' is shown, followed by location and creator information: 'Edificio A, Planta 1, PUESTO 21' and 'Creada por: Aitor Gómez'. A search bar labeled 'Organizador' contains the text 'user'. Below the search bar is a table listing users.

Organizador	
User Demo9	userdemo9@demo.com
User Demo8	userdemo8@demo.com
User Demo5	userdemo5@demo.com
User Demo10	userdemo10@demo.com
User Demo2	userdemo2@demo.com
User Demo4	userdemo4@demo.com
User Demo3	userdemo3@demo.com
User Demo1	userdemo1@demo.com
User Demo6	userdemo6@demo.com
User Demo7	userdemo7@demo.com

Figura 23. Diálogo de conflictividad al crear reserva de puesto de trabajo.

The screenshot displays a web application interface for creating a reservation. On the left, a sidebar contains the 'a l o g o' logo and a 'Company' section with a 'Reservas' link. The main content area is titled 'Reserva de PUESTO 21' and shows details for 'Edificio A, Planta 1, PUESTO 21', created by 'Aitor Gómez'. It includes input fields for 'Organizador' (John Doe), 'Fecha' (5/7/2023), 'Hora de inicio' (11:45), and 'Hora de fin' (15:45). At the bottom of this form are 'Reservar' and 'Cancelar' buttons. A modal dialog titled 'Reservas conflictivas' is open in the center, asking if the user wants to continue with the reservation despite conflicts. It contains a table with one row showing a conflict for the time '10:15 - 23:45' by user 'John Doe'. The dialog has 'Atras' and 'Continuar' buttons at the bottom right.

Organization / Reservas / Nueva reserva de puesto

Reserva de PUESTO 21

📍 Edificio A, Planta 1, PUESTO 21

👤 Creada por: Aitor Gómez

Organizador
John Doe

Fecha
5/7/2023

Hora de inicio
11:45

Hora de fin
15:45

Reservar Cancelar

Reservas conflictivas

Una o más reservas coinciden total o parcialmente con la que estás creando. ¿Deseas continuar con tu reserva anulando las coincidentes?

Hora	Usuario
10:15 - 23:45	John Doe

Atras Continuar

4.3.6. Ficha de reserva de puesto de trabajo

Una vez el usuario crea la reserva de puesto de trabajo, es redirigido a la vista de ficha de reserva, donde se muestra información de ésta y se le permiten realizar diferentes acciones sobre ésta.

En la URL está presente el identificador de la reserva que acaba de ser creada por el usuario, y de esta forma el sistema es capaz de buscar los datos de la reserva en la base de datos.

Desde esta vista el usuario puede visualizar lo siguiente:

- El usuario organizador de la reserva.
- El tipo de recurso (si es puesto de trabajo o sala de reuniones) reservado, así como el nombre del recurso.
- La fecha, hora de inicio y hora de fin de la reserva.
- La planta y edificio donde se encuentra el recurso reservado.

Estos valores no son editables desde esta vista, ya que para editar una reserva es necesario acceder a la vista dedicada a esta funcionalidad.

El usuario puede, desde el botón dedicado para ello, realizar tanto el check in como el check out, acceder a la vista de edición de reserva o anular la reserva.

En caso de que el usuario anule la reserva, será redirigido a la vista inicial de búsqueda.

Si el usuario desea volver a la vista del listado de recursos, debe pulsar en el botón “Volver”.

El XHTML asociado a esta vista es **workstation_booking.xhtml**, que está conectado con el bean **WorkstationBookingBean** y que a su vez utiliza varios servicios:

- **WorkstationBookingService**, utilizado para recuperar la reserva de puesto de trabajo y su información.
- **ResourceService**, utilizado para recuperar la información de la planta y el edificio al que pertenece el recurso reservado, y así mostrar su información.
- **BookingService**, utilizado para realizar las acciones de check in, check out y anular reserva.

Figura 24. Ficha de reserva de puesto de trabajo.

The screenshot displays the 'Ficha de reserva' (Reservation Card) interface. On the left, a sidebar contains the 'a l o g o' logo and navigation links for 'Company' and 'Reservas'. The main content area features the title 'Ficha de reserva' and a breadcrumb trail: 'Organization / Reservas / Ficha de reserva de espacio'. The reservation details are as follows:

Fecha	Hora de inicio	Hora de fin
4/7/2023	13:45	14:45

Ubicación: Edificio A, Planta 1, PUESTO 21

User profile: User Demo9 (with a dropdown menu containing: Check in, Check out, Editar, Anular reserva)

Puesto de trabajo: PUESTO 21

A 'Volver' button is located at the bottom right of the reservation details.

4.3.7. Edición de reserva de puesto de trabajo

Esta vista puede ser accedida desde dos puntos de la aplicación:

- Desde el listado de reservas del panel lateral que aparece al pulsar en un puesto de trabajo en el listado de recursos.
- Desde la opción de editar reserva dentro de la ficha de reserva de puesto de trabajo.

Una vez el usuario accede a esta vista, puede modificar la fecha y hora de inicio y fin de la reserva. El proceso de edición, al igual que el proceso de creación, comprueba que tanto el comienzo como el fin de la reserva son válidos (que la fecha y hora de inicio no sea anterior a la actual, y que la hora de fin no sea anterior a la hora de inicio).

Además, el sistema también comprueba si el recurso tiene reservas que sean conflictivas con el horario introducido por el usuario. Si el recurso tiene reservas conflictivas, el sistema muestra un diálogo indicando que para proceder con la edición de reserva el usuario debe anular las reservas existentes.

Una vez el usuario edite la reserva, puede volver a la vista del listado de recursos pulsando en el botón “Volver”.

El XHTML asociado a esta vista es **edit_workstation.xhtml**, siendo **EditWorkstationBean** el bean con el que se conecta.

Este bean utiliza varios servicios:

- **WorkstationBookingService**, para recoger información de la reserva de puesto de trabajo.
- **EditWorkstationBookingService**, encargado de procesar la edición de la reserva en la base de datos.
- **ResourceService**, para recoger la información del recurso reservado, así como la planta y el edificio al que pertenece.
- **BookingService**, utilizado para buscar reservas conflictivas sobre ese recurso para fecha y hora de inicio y fin seleccionadas por el usuario.

Figura 25. Edición de reserva de puesto de trabajo.

The screenshot shows a web application interface for editing a reservation. On the left is a vertical sidebar with the 'a l o g o' logo and a 'Company' section containing a 'Reservas' link. The main content area has a breadcrumb trail 'Organization / Reservas / Editar reserva de puesto' and a title 'Reserva de PUESTO 21'. Below the title, it displays reservation details: 'Edificio A, Planta 1, PUESTO 21', 'Creada por: Aitor Gómez', and 'Organizada por: User Demo9'. There are three input fields: a date field 'Fecha' with the value '4/7/2023', an 'Hora de inicio' field with '13:45', and an 'Hora de fin' field with '14:45'. At the bottom of the form are two buttons: 'Editar reserva' (highlighted in blue) and 'Volver' (grey). A share icon is located in the top right corner of the page.

4.3.8. Creación de reserva de sala de reuniones

Si el usuario pulsa en el botón de “Nueva reserva” desde el menú lateral de una sala de reuniones, se le redirige a la vista de crear reserva de sala de reuniones.

De la misma forma que para el resto de las vistas, es necesario comunicar mediante parámetros URL la fecha y hora de inicio, la fecha y hora de fin, así como el identificador del recurso seleccionado. Para lograr esto, se utilizan los siguientes parámetros:

- **start:** Fecha y hora de inicio de la reserva en milisegundos.
- **end:** Fecha y hora de fin de la reserva en milisegundos.
- **resourceId:** Identificador del recurso sobre el que se va a realizar la reserva.

Desde esta vista el usuario puede modificar la fecha y la hora de la reserva respetando los límites mínimos de fecha, siendo este límite la fecha actual. También es posible modificar la hora, no permitiendo completar la reserva si la fecha y hora de inicio es menor a la actual a la hora de completar el proceso de reserva.

Ya que la aplicación está orientada a crear reservas a terceros (el caso de uso principal sería el personal de secretaría de una compañía crea reservas al resto de empleados) es necesario seleccionar un organizador. Este organizador será un usuario registrado en el sistema y que pertenezca a la misma organización que el usuario con sesión iniciada.

Para seleccionar un organizador, el usuario debe escribir en el campo de texto un mínimo de tres caracteres para que el sistema realice la consulta apropiada a la base de datos. Esta consulta busca usuarios cuyo nombre completo (nombre + apellidos) o email contiene la cadena de texto introducida por el usuario. Si el sistema encuentra resultados, devuelve un listado de usuarios que aparece debajo el componente de forma flotante, pudiendo el usuario seleccionar cualquiera de ellos.

En este tipo de reserva, el usuario también puede incluir un título a la reserva de máximo 255 caracteres, así como una descripción de esta. El componente utilizado para la descripción de la reserva permite enriquecer el texto utilizado negrita, cursiva, subrayado y tachado.

Al ser una reserva de sala de reuniones el usuario puede añadir invitados a la reunión, que son usuarios que pertenecen a la organización del invitado. El componente utilizado a la hora de buscar usuarios para añadir como invitados tiene el mismo funcionamiento que el

componente utilizado para seleccionar un organizador, siendo la única diferencia que la consulta a la base de datos no devolverá los usuarios que ya hayan sido añadidos como invitados a la reserva y tampoco devolverá al usuario organizador seleccionado como posible invitado.

Si el usuario añade un invitado no podrá añadir a ese mismo usuario como organizador de la reserva, debiendo eliminar primero al usuario como invitado y posteriormente añadirlo como organizador.

Cuando se añade un invitado, éste aparece en una tabla situada debajo del componente que permite buscar usuarios para añadirlos como invitados, siendo posible eliminarlos de la tabla, es decir, eliminar a ese usuario invitado de la reserva, así como marcarlo opcional, significando esto que su asistencia no es requerida. Por defecto, la asistencia de un usuario a una reserva es requerida.

Al pulsar el botón “Reservar”, el sistema realiza varias comprobaciones:

- Si el usuario no ha seleccionado un organizador para la reserva, se le muestra un mensaje de error impidiendo completar el proceso.
- Si la fecha u hora son inválidas (fecha y hora de inicio anterior a la actual u hora de fin anterior a hora de inicio) se le muestra un mensaje de error al usuario indicándolo.
- Se buscan las posibles reservas del recurso a reservar que coincidan total o parcialmente con la fecha y hora seleccionada por el usuario. En caso de que exista, se le muestra un diálogo al usuario indicando que continuar con el proceso anulará las reservas existentes (un recurso no puede tener dos reservas simultáneas que se solapen en el tiempo). Si el usuario decide proceder, se anulan las reservas existentes sobre ese recurso antes de crear la nueva reserva.

Si todas las comprobaciones son exitosas, la reserva es creada por el sistema y el usuario es redirigido a la ficha de reserva de sala de reuniones.

Figura 26. Creación de reserva de sala de reuniones.

The screenshot shows the 'Reserva de SALA 4' form in the Alogo web application. The interface includes a sidebar with the 'Company' logo and a 'Reservas' menu item. The main content area displays the reservation details for 'SALA 4'.

Reserva de SALA 4

📍 Edificio A, Planta 1 SALA 4
👤 Creada por: Aitor Gómez

Organizador:
User Demo5

Título:
Reunión semanal

Descripción:
B I U S
Objetivo: Debater el progreso ocurrido a lo largo de la semana

Fecha:
5/7/2023

Hora de inicio:
11:15

Hora de fin:
12:15

Añadir invitados

Invitados:
Buscar usuario

Opcional	Nombre	Email	
<input type="checkbox"/>	John Doe	john.doe@demo.com	🗑️
<input checked="" type="checkbox"/>	User Demo10	userdemo10@demo.com	🗑️

Reservar **Cancelar**

Figura 27. Diálogo de conflictividad al crear reserva de sala de reuniones.

The screenshot displays a web application interface for room reservations. On the left, a sidebar contains the 'a l o g o' logo and a 'Company' section with a 'Reservas' link. The main content area is titled 'Reserva de SALA 4' and shows details for 'Edificio A, Planta 1, SALA 4', created by 'Aitor Gómez'. The form includes fields for 'Organizador' (User Demo3), 'Titulo' (Test reserva de espacio), and a rich text 'Descripción' area. Below these are fields for 'Fecha' (5/7/2023), 'Hora de inicio' (10:45), and 'Hora de fin' (12:45). An 'Añadir invitados' section features a search bar and a table of invited users. At the bottom are 'Reservar' and 'Cancelar' buttons. A modal dialog titled 'Reservas conflictivas' is open in the center, asking if the user wants to continue with the reservation despite conflicts. It lists a conflict for '11:15 - 12:15' by 'User Demo5' and provides 'Atras' and 'Continuar' buttons.

Reserva de SALA 4

📍 Edificio A, Planta 1, SALA 4
👤 Creada por: Aitor Gómez

Organizador
User Demo3

Título
Test reserva de espacio

Descripción
B I U S

Fecha
5/7/2023

Hora de inicio
10:45

Hora de fin
12:45

Añadir invitados

Invitados
Buscar usuario

Opcional	Nombre	Email
<input checked="" type="checkbox"/>	User Demo3	user.demo3@demo.com

Reservar Cancelar

Reservas conflictivas

Una o más reservas coinciden total o parcialmente con la que estás creando. ¿Deseas continuar con tu reserva anulando las coincidentes?

Hora	Usuario
11:15 - 12:15	User Demo5

Atras Continuar

4.3.9. Ficha de reserva de sala de reuniones

Una vez el usuario crea la reserva de puesto de trabajo, es redirigido a la vista de ficha de reserva, donde se muestra información de ésta y se le permiten realizar diferentes acciones sobre ésta.

En la URL está presente el identificador de la reserva que acaba de ser creada por el usuario, y de esta forma el sistema es capaz de buscar los datos de la reserva en la base de datos.

Desde esta vista el usuario puede visualizar lo siguiente:

- El usuario organizador de la reserva.
- El tipo de recurso (si es puesto de trabajo o sala de reuniones) reservado, así como el nombre del recurso.
- La fecha, hora de inicio y hora de fin de la reserva.
- La planta y edificio donde se encuentra el recurso reservado.
- El título y descripción de la reserva, si se introdujeron en el proceso de creación de reserva de sala de reuniones.
- El listado de invitados presentes en la reserva, así como si son requeridos u opcionales.

Estos valores no son editables desde esta vista, ya que para editar una reserva es necesario acceder a la vista dedicada a esta funcionalidad.

El usuario puede, desde el botón dedicado para ello, realizar tanto el check in como el check out, acceder a la vista de edición de reserva o anular la reserva.


En caso de que el usuario anule la reserva, es redirigido a la vista inicial de búsqueda.

Si el usuario desea volver a la vista del listado de recursos, debe pulsar en el botón “Volver”.

El XHTML asociado a esta vista es **space_booking.xhtml**, siendo **SpaceBookingBean** el bean conectado a dicha vista. Este bean utiliza los siguientes servicios:

- **SpaceBookingService**, para recoger la información de la reserva de sala de reuniones.
- **ResourceService**, para recoger la información del recurso, así como la del edificio y planta a los que pertenece.
- **BookingService**, para realizar las acciones de check in, check out y anulación de reserva.


Figura 28. Ficha de reserva de sala de reuniones.



Company
Reservas

Organization / Reservas / Ficha de reserva de espacio

Ficha de reserva



User Demo5

Sala de reuniones
SALA 4

Reunión semanal
Objetivo: Debatar el progreso ocurrido a lo largo de la semana

Fecha
6/7/2023

Hora de inicio
11:15

Hora de fin
12:15

Ubicación
Edificio A, Planta 1, SALA 4

2 invitados

Opcional	Nombre	Email
<input checked="" type="checkbox"/>	User Demo10	user.demo10@demo.com
<input type="checkbox"/>	John Doe	john.doe@demo.com

Volver

Figura 29. Opciones sobre la reserva en ficha de reserva de sala de reuniones.


Logo

Company

Reservas


Organization / Reservas / Ficha de reserva de espacio

Ficha de reserva




Fecha: 5/7/2023 Hora de inicio: 11:15 Hora de fin: 12:15

Ubicación: Edificio A, Planta 1, SALA 4

 User Demo5

- Sala de reuniones
- SALA 4**
- Reunión semanal
- Objetivo: Debater el progreso ocurrido a lo largo de la semana



- ✓ Check in
- ✗ Check out
- ✎ Editar
- 🗑 Anular reserva

Asistentes

Personal	Nombre	Email
	User Demo10	userdemo10@demo.com
	John Doe	john.doe@demo.com

Volver

4.3.10. Edición de reserva de sala de reuniones

Esta vista puede ser accedida desde dos puntos de la aplicación:

- Desde el listado de reservas del panel lateral que aparece al pulsar en una sala de reuniones en el listado de recursos.
- Desde la opción de editar reserva dentro de la ficha de reserva de sala de reuniones.

Una vez el usuario accede a esta vista, puede modificar:

- Título y descripción de la reserva.
- La fecha y hora de inicio y fin.
- Los invitados de la reserva, pudiendo eliminar y añadir nuevos.

El proceso de edición, al igual que el proceso de creación, comprueba que tanto el comienzo como el fin de la reserva sean válidos (que la fecha y hora de inicio no sea anterior a la actual, y que la hora de fin no sea anterior a la hora de inicio).

Además, el sistema también comprueba si el recurso tiene reservas que sean conflictivas con el horario introducido por el usuario. Si el recurso tiene reservas conflictivas, el sistema muestra un diálogo indicando que para proceder con la edición de reserva el usuario debe anular las reservas existentes.

Una vez el usuario edite la reserva, puede volver a la vista del listado de recursos pulsando en el botón “Volver”.

El XHTML asociado a esta vista es **edit_space.xhtml**, y el **bean** con el que se conecta es **EditSpaceBookingBean**. Este **bean** también utiliza el bean de sesión **SessionBean**, ya que necesita saber el identificador de la organización a la que pertenece el usuario que ha iniciado sesión para la búsqueda de usuarios en base de datos. Los servicios utilizados por el bean **EditSpaceBookingBean** son:

- **SpaceBookingService**, que se encarga de recoger la información de la reserva de sala de reuniones.
- **EditSpaceService**, encargado de progresar la edición de la reserva a la base de datos.
- **ResourceService**, para recuperar la información de la planta y el edificio al que pertenece el recurso.

- **BookingService**, encargado de buscar reservas conflictivas para la fecha y hora de inicio y fin introducidas por el usuario.
- **UserService**, para buscar usuarios en la base de datos que coincidan con la cadena introducida por el usuario en el buscador de invitados.

Figura 30. Edición de reserva de sala de reuniones.

a l o g o

Company

Reservas

Organization / Reservas / Editar reserva de espacio

Reserva de SALA 4

📍 Edificio A, Planta 1, SALA 4

👤 Creada por: Aitor Gómez

👤 Organizada por: User Demo5

Título
Reunión semanal

Descripción
B I U G

Objetivo: Debater el progreso ocurrido a lo largo de la semana y comentar el próximo roadmap de la compañía.

Fecha
15/7/2023

Hora de inicio
11:15

Hora de fin
12:15

Invitados
Buscar usuario

Opcional	Nombre	Email	
<input checked="" type="checkbox"/>	User Demo1	user.demo1@demo.com	
<input type="checkbox"/>	John Doe	john.doe@demo.com	
<input checked="" type="checkbox"/>	User Demo10	user.demo10@demo.com	

Editar reserva Volver

4.4. Evaluación de resultados

Para comprobar que la aplicación funciona correctamente y cumplimenta los requisitos y objetivos planteados inicialmente, se ha realizado una fase de pruebas extensa que recorre todas las funcionalidades de la aplicación, buscando cualquier tipo de fallo en la lógica o comportamiento extraño que podría ser confuso para el usuario.

También se ha hecho hincapié en la usabilidad del producto y por tanto buscando que la aplicación web responda de forma rápida y fluida en todo momento, para que el usuario se sienta cómodo usando la aplicación.

Además, se han hecho pruebas de la aplicación en varias resoluciones, así como emulando dispositivos con resoluciones variadas para comprobar que la aplicación escalase correctamente en el mayor número de resoluciones posibles.

Esto se ha logrado gracias a las herramientas de desarrollador que Google Chrome pone en nuestra mano, donde podemos elegir diferentes resoluciones, así como dispositivos a emular. Como ejemplo, se añade una captura de pantalla usando dichas herramientas de Google Chrome y una resolución de 1600 píxeles de ancho por 900 píxeles de alto, que es una resolución habitual en ordenadores portátiles.


Figura 31. Vista del listado de recursos a una resolución de 1600x900.

Dimensions: Responsive ▾

1600 × 900

100% ▾

No throttling ▾



Company

Reservas

Organization / Reservas / Listado

Reservas por recurso

Selecciona una fecha y ubicación para visualizar los recursos disponibles y gestionar sus reservas.

Edificio

Edificio A ▾

Planta

Planta 1 ▾

Fecha

05-07-2023

Buscar recurso

Mostrar filtros

Ocupación desde 05-07-2023 - 12:15 hasta 05-07-2023 - 23:45

Libre 89%

Totalmente ocupado 3%

Parcialmente ocupado 7%

	Nombre ↑↓	Capacidad ↑↓	Edificio	Planta	Categoría
	<div></div> PUESTO 21	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
	<div></div> SALA 4	20 persona(s)	Edificio A	Planta 1	Sala de reuniones
	<div></div> SALA 10	8 persona(s)	Edificio A	Planta 1	Sala de reuniones
	<div></div> SALA 17	30 persona(s)	Edificio A	Planta 1	Sala de reuniones
	<div></div> PUESTO 9	1 persona(s)	Edificio A	Planta 1	Puesto de trabajo
	<div></div> SALA 12	12 persona(s)	Edificio A	Planta 1	Sala de reuniones

5. Conclusiones y trabajo futuro

En este capítulo se disponen las conclusiones con el trabajo, y en específico si se han logrado todos los objetivos planteados inicialmente. También se indican las posibles mejoras y funcionalidades añadidas de las que podría beneficiarse la aplicación, otorgando una mayor flexibilidad a los usuarios finales.

5.1. Conclusiones del trabajo

En el apartado 3.2 se listaron los diferentes objetivos específicos que se esperaban cumplir al dar por finalizado el desarrollo de la aplicación.

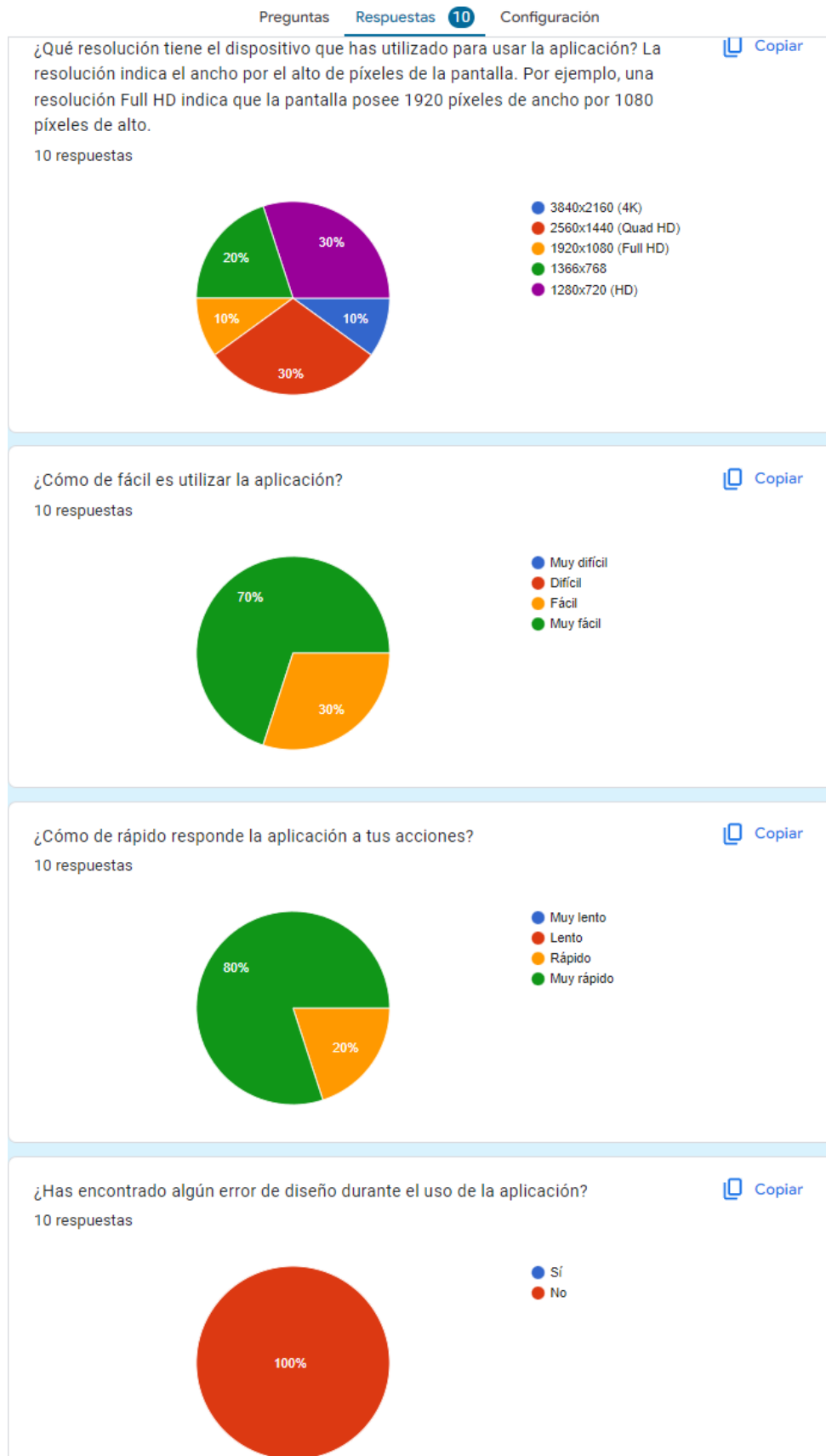
En cuanto al objetivo “Implementación de una aplicación web que permita gestionar reservas”, podemos afirmar que se ha cumplido ya que la aplicación web permite realizar reservas, editarlas y anularlas, tal y como se ha dispuesto en el apartado cuatro, específicamente en el apartado 4.1, donde se disponen de forma detallada la funcionalidad de cada vista de la aplicación.

El objetivo “Capacidad para realizar la confirmación de asistencia en nombre de un empleado (check-in)”, que suponía un objetivo importante a cumplir porque nos diferencia en gran medida del resto de aplicaciones presentes actualmente en el mercado, ha sido cumplido tal y como se ha demostrado en el apartado 4.2.4.2, 4.2.6 y 4.2.9.

Se ha conseguido cumplir también el tercer objetivo, “poder observar los recursos y su ocupación para una fecha y hora concretas”, tal y como se demuestra en el apartado 4.2.4.

Para comprobar que la usabilidad de la aplicación es adecuada, diez usuarios externos al desarrollo de la aplicación han usado la aplicación durante un tiempo determinado. Una vez el tiempo de uso ha acabado, se les ha indicado realizar un formulario de Google donde se les solicita que respondan a diferentes cuestiones relacionadas con la aplicación. A continuación, se ilustra mediante una captura los resultados de dicho formulario.

Figura 32. Resultados del formulario de usabilidad



Como se puede observar en los resultados, el 70% han respondido que la facilidad de uso de la aplicación es “Muy fácil”, y el 30% restante han respondido “Fácil”. De la misma forma, para la pregunta relacionada con el rendimiento de la aplicación, el 80% de los usuarios han respondido “Muy rápido”, mientras que el otro 20% ha respondido “Rápido”.

Teniendo en los resultados obtenidos, podemos concluir que hemos cumplido de usabilidad expuesto en el apartado 3.2.

Para mantener cierta organización en las tareas a realizar, se ha utilizado una metodología basada en SPRINTS (pero modificada ya que en este desarrollo no se cuenta con las figuras de SCRUM Master ni con un equipo de desarrollo compuesto por varias personas).

Cada semana, empezando desde el 15 de mayo de 2.023, y hasta la semana del 26 de junio de 2.023 se han propuesto diversas tareas que debían ser completadas en esa misma semana. Estas tarjetas pasaban de estar en la columna “Sin empezar” a la columna “Activas”, y una vez eran completadas, se trasladaban a la columna “Terminadas”.

Además, gracias a utilizar la funcionalidad de fecha de vencimiento de una tarjeta, es posible saber si ha ocurrido algún retraso o imprevisto durante el desarrollo, ya que en el caso de que no se complete una funcionalidad a tiempo el propio Trello marcará la tarjeta con la etiqueta de “Plazo vencido”.

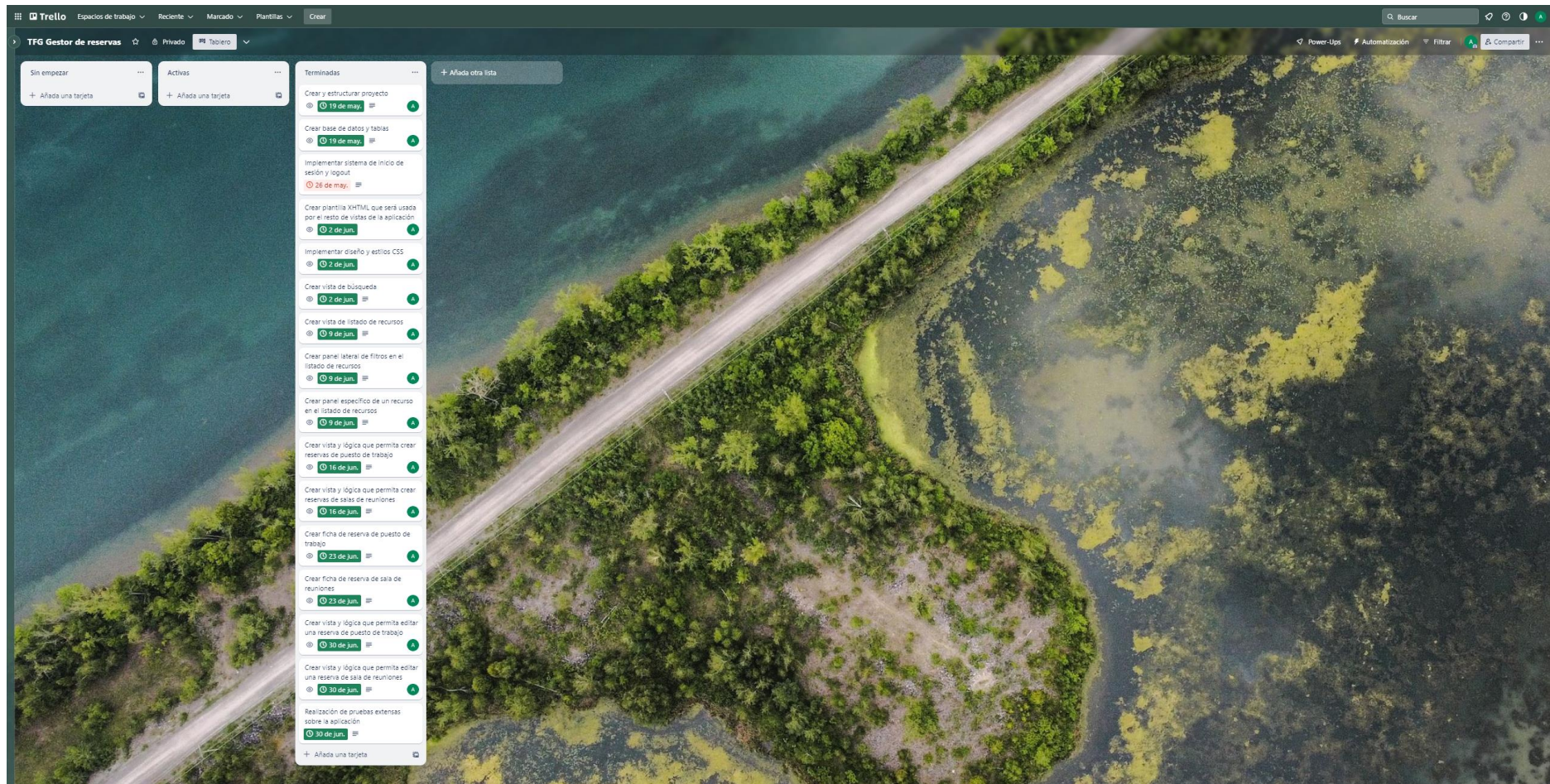
Durante el desarrollo de la aplicación, solo hubo plazos vencidos para la tarjeta relacionada con la implementación del sistema de inicio sesión mediante Spring Security, ya que esta funcionalidad supuso unos retos inesperados.

En base a lo enunciado en este capítulo, y el haber cumplido los objetivos individuales dispuestos en el apartado 3.2, podemos concluir que el objetivo general de la aplicación web ha sido cumplido.

Para validar el funcionamiento de la aplicación, se realizaron una serie de pruebas recorriendo el flujo normal de la aplicación, así como pruebas específicas para comprobar si la lógica de la aplicación contenía fallos.

Como último apunte personal, quiero destacar el gran aprendizaje obtenido gracias a la realización de este trabajo, tanto a nivel de conocimientos de programación como la mejora en la capacidad de organización, ya que todo depende de uno mismo.

Figura 33. Tablero de Trello al acabar el desarrollo de la aplicación.



5.2. Líneas de trabajo futuro

Aunque se hayan completado todos los objetivos propuestos inicialmente, la aplicación puede recibir una serie de funcionalidades extra que le otorgarían mayor potencia y una capacidad mucho más notoria para competir contra otras soluciones ya existentes en el mercado.

Estas posibles funcionalidades son:

- Añadir los planos de cada planta y registrar las coordenadas de cada recurso dentro de la planta, para luego poder mostrar los recursos sobre el plano, y de esta forma ofrecer una alternativa al listado de recursos mucho más visual.
- Capacidad de recurrencia a la hora de crear reservas, es decir, que la aplicación permita reservar para más de un día a la vez. Esta funcionalidad reduciría la cantidad de acciones necesarias por el usuario y, por tanto, el tiempo que necesita para crear reservas en la aplicación.
- Sincronización con calendarios externos como Google o Outlook, ya que estos dos sistemas poseen APIs para lograr esta sincronización.
- Añadir invitados externos a la organización, ya que una compañía puede que invite a una reunión a personas que no trabajan en su compañía, y por tanto no están registrados como tal en el sistema.
- Añadir la capacidad de reservas otros tipos de recursos, como plazas de parking o mesas de comedores.

Referencias bibliográficas

- Adams, G., Borges, B., & Erickson, K. (2023, abril 4). *Motivos para pasar a Java 11 y versiones posteriores.*
- Alarcón, J. M. (2021, marzo 23). *La API de persistencia de Java: ¿Qué es JPA?*
- Banks, C. (2020, abril 30). *imac plateado sobre mesa de madera marrón.*
- Condeco. (s. f.). *Software de reserva de escritorio.*
- Creative Commons. (s. f.). *CC0 License.*
- Deaner, D. (2018, febrero 1). *mesa de conferencia de madera beige.*
- Google. (s. f.). *Add a room to an event.*
- Kadence WP. (s. f.). *Free Placeholder Logos.*
- Kalena. (s. f.). *¿Cómo funciona Kalena?*
- MySQL. (s. f.). *MySQL Workbench: SQL Development.*
- Natchimuthu, A. (2022, julio 18). *Why Trello is great for project management: 4 Tips to make the best out of Trello.*
- PrimeTek. (s. f.-a). *PrimeFaces Showcase.*
- PrimeTek. (s. f.-b). *PrimeFaces Theme Store.*
- Spring. (s. f.). *Spring Data JPA.*
- Spring Framework. (s. f.). *Spring Boot.*
- Unsplash. (s. f.). *Licencia Unsplash.*
- Wikipedia Authors. (s. f.). *JavaServer Faces.*
- Yanina Muradas. (2018, junio 5). *Qué es Spring Framework y por qué usarlo.*

Índice de acrónimos

A

API: Application Programming Interface.

C

CRUD: Create, Read, Update and Delete.

CSS: Cascading Style Sheets.

J

JPA: Java Persistence API.

JSF: JavaServer Faces.

U

UUID: Universally Unique Identifier.

URL: Uniform Resource Locator.

X

XHTML: eXtensible HyperText Markup Language.