



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Aplicación de Gestión de  
Reserva de Aulas desarrollada  
con Vaadin y Spring Boot**



Presentado por Lisa Cané Sáiz  
en Universidad de Burgos — 20 de julio  
de 2020

Tutor: Jesús Manuel Maudes Raedo







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



Dr. D. Jesús Manuel Maudes Raedo, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que la alumna D. Lisa Cané Sáiz, con DNI 71306263-F, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado *Aplicación de Gestión de Reserva de Aulas desarrollada con Vaadin y Spring Boot*.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 20 de julio de 2020

Vº. Bº. del Tutor:

D. Jesús Manuel Maudes Raedo





## Resumen

La reserva de aulas y salas, ya sea para impartir clases, exámenes, cursos o reuniones, es el *pan de cada día* de los centros educativos. Por ello, su correcta gestión es importante si se quieren evitar reservas concurrentes del mismo aula, extravío o pérdida de las mismas.

La digitalización de este proceso de reserva puede solucionar muchos de estos problemas, ya que es la propia persona interesada quien hace la reserva, o un encargado. De esta forma se disminuye el problema de introducir datos erróneos, ya sea que se limita la posibilidad de malentendidos y se hace un control de los datos introducidos a través del software.

Por lo tanto, este proyecto ha tomado forma de aplicación web de gestión de reservas, desarrollándose con Vaadin y Spring Boot.

## Descriptores

Gestión de reservas, aplicación web, Vaadin, Spring Boot.

### **Abstract**

Scheduling classrooms reservations, whether to teach classes, exams, courses or meetings, its very common in educational centers. Therefore, its correct management is important if the aim is to avoid concurrent reservations of the same classroom, or the loss of this reservations.

The digitization of this reservation process can solve many of these problems, since the interested person is the one who makes the reservation, or a manager. This reduces the problem of entering erroneous data, since it limits the possibility of misunderstanding and controls the data entered through the software.

Therefore, this project has taken the form of a reservation management web application, developed with Vaadin and Spring Boot.

### **Keywords**

Scheduling reservations, web application, Vaadin, Spring Boot.



---

# Índice general

---

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales . . . . .	3
2.2. Objetivos técnicos . . . . .	3
Conceptos teóricos	5
3.1. <i>Framework</i> . . . . .	5
3.2. <i>Frontend</i> . . . . .	5
3.3. <i>Backend</i> . . . . .	5
3.4. Desarrollo <i>full stack</i> . . . . .	6
3.5. Convención sobre configuración . . . . .	6
Técnicas y herramientas	7
4.1. Código fuente . . . . .	7
4.2. Eclipse . . . . .	9
4.3. Base de datos . . . . .	9
4.4. Gestión del proyecto y control de versiones . . . . .	9
4.5. Overleaf . . . . .	10
4.6. <i>Hosting</i> para despliegue . . . . .	10
4.7. Herramientas de diseño . . . . .	11

<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>13</b>
5.1. Motivación de la elección del proyecto . . . . .	13
5.2. Periodo de prueba y toma de contacto más largo de lo esperado	13
5.3. Vaadin . . . . .	14
5.4. Cambio de versión de Vaadin y fracaso en la integración de las APIs de Google . . . . .	14
5.5. Spring Security . . . . .	15
5.6. Spring Data JPA . . . . .	16
5.7. Uso del interfaz <i>Specification</i> de Spring Data JPA para las consultas . . . . .	16
5.8. Consulta de disponibilidad de aulas . . . . .	16
<b>Trabajos relacionados</b>	<b>17</b>
6.1. Sistema Interno de Reserva de Espacios de la EPS de la Universidad de Sevilla . . . . .	17
6.2. Booked . . . . .	18
6.3. SuperSaaS . . . . .	18
6.4. Acuity Scheduling . . . . .	18
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>19</b>
7.1. Conclusiones . . . . .	19
7.2. Líneas de trabajo futuras . . . . .	20
<b>Bibliografía</b>	<b>21</b>

---

## Índice de figuras

---

---

## Índice de tablas

---

---

# Introducción

---

Con fin de solventar los problemas que tienen los profesores a la hora de reservar aulas para la realización de exámenes, o ver si se encuentran disponibles, sin necesidad de desplazarse o consultar papeles que puede que no estén al día, ha surgido este proyecto centrado en desarrollar aplicación web para llevar a cabo su gestión.

Lo que se pretende con el mismo es permitir a cualquier usuario consultar tanto las reservas registradas como la disponibilidad de las aulas. De esta forma, un profesor que esté planificando su examen puede ver qué horario y aula le convienen, consultando si está disponible en una fecha y hora específicas o filtrando a través de ciertos parámetros (capacidad, número de ordenadores, centro en el que se encuentra, etc.).

Una vez haya decidido la fecha, hora y lugar de la reserva, acudirá al responsable del departamento que se encuentra a cargo del aula, o en su defecto al responsable del centro, para pedirle que la registre. Si por algún casual esta reserva se tiene que modificar, o incluso eliminar, deberá acudir de nuevo a él.



---

# Objetivos del proyecto

---

En este apartado se detallan los objetivos perseguidos con la realización del proyecto.

## 2.1. Objetivos generales

- Desarrollar una aplicación web para la gestión de reservas de aulas de los centros educativos.
- Facilitar la consulta de las reservas realizadas y de la disponibilidad de las aulas de los centros a cualquier persona.
- Permitir al administrador gestionar los centros, departamentos, aulas y usuarios registrados en el sistema.
- Permitir al administrador consultar el histórico con las operaciones que se han realizado sobre las reservas (creaciones, modificaciones y eliminaciones).
- Orientar la aplicación a los espacios de la universidad, teniendo en cuenta detalles como la capacidad del aula y el número de ordenadores.

## 2.2. Objetivos técnicos

- Desarrollar una aplicación web de tipo cliente servidor con Vaadin y Spring Boot.

- Brindar la máxima seguridad integrando tecnologías como Spring Security, restringiendo el acceso a usuarios no autorizados a ciertas funcionalidades.
- Utilizar un repositorio de GitHub para llevar a cabo el control de versiones.
- Estructurar correctamente los datos con los que se trabaja, creando un modelo relacional lo más normalizado posible.



---

# Conceptos teóricos

---

En este apartado se explican diferentes conceptos teóricos relacionados con el proyecto.

## 3.1. *Framework*

Un *framework* [15], o marco de trabajo, es una herramienta software que proporciona una base de código y formas, estandarizadas y consistentes para el desarrollo de aplicaciones web.

## 3.2. *Frontend*

El *frontend* [4] es con lo que el usuario interactúa en una web, es la parte a la que se tiene acceso directamente.

Los lenguajes de desarrollo de *frontend* más comunes son HTML, CSS y JavaScript, aunque también se puede hacer uso de frameworks y librerías como React, Angular, Bootstrap ...

## 3.3. *Backend*

El *backend* [4] es la capa de acceso a los datos, que se conecta con la base de datos y el servidor utilizados en la web, y que no es accesible por los usuarios, ya que contiene la parte lógica de la aplicación.

Los lenguajes de desarrollo de *backend* más comunes son Java, Python, Ruby ...

### 3.4. Desarrollo *full stack*

El desarrollo *full stack* [3] es aquel que engloba tanto la producción del lado del cliente o *frontend* como la del lado del servidor o *backend*.

### 3.5. Convención sobre configuración

La *Convención sobre Configuración* [26] (CoC) es un paradigma de programación software con el cual se tratan de minimizar las decisiones que debe tomar el desarrollador, simplificando sin perder flexibilidad.

---

# Técnicas y herramientas

---

En este apartado se presentan las tecnologías y herramientas utilizadas para llevar a cabo el desarrollo del proyecto.

El criterio de elección para la mayoría de ellas ha sido el previo conocimiento y trabajo con ellas a lo largo de la carrera, como es el caso de Java, Eclipse, PostgreSQL junto a pgAdmin, GitHub junto a ZenHub y GitHub Desktop, StarUML y Balsamiq Wireframes.

## 4.1. Código fuente

### Java

Java [14] es un lenguaje de programación orientado a objetos para desarrollo de aplicaciones.

### JavaScript

JavaScript [12] es un lenguaje de programación interpretado, que permite el desarrollo del *frontend* de aplicaciones y sitios web.

### CSS

CSS (*Cascading Style Sheets* u Hojas de Estilo en Cascada) [25] es un lenguaje declarativo que permite cambiar la apariencia de los componentes de HTML.

## Vaadin

Vaadin [22] es una plataforma de código abierto (*open source*) para el desarrollo de aplicaciones web con Java, permitiendo también el uso de HTML, JavaScript, CSS, entre otros.

Los pre-requisitos para poder utilizar Vaadin son tener instalado JDK 8 (*Java Development Kit*) y un IDE (Entorno de Desarrollo Integrado) de Java como Eclipse, NetBeans o IntelliJ Idea. En caso de utilizar Eclipse, es necesario instalar *plugin* de Vaadin.

## Spring Boot

Spring Boot [23] es una solución o simplificación del *framework* Spring de Java. Sigue el paradigma [Convención sobre configuración](#), proporcionando una estructura básica configurada del proyecto e incluyendo directamente las bibliotecas de terceros necesarias.

Los pre-requisitos para poder utilizar Spring Boot son tener instalado como mínimo JDK 8 (*Java Development Kit*), la versión actualizada de Spring, para compilar se puede utilizar Maven o Gradle y se puede incorporar un *servlet* Apache Tomcat, Jetty o Undertow.

La combinación de Vaadin y Spring Boot ha facilitado mucho el desarrollo, ya que este último permite centrarse en el desarrollo de código sin tener que configurar el servidor o las dependencias, simplemente se incluyen las “importaciones” en el archivo *pom.xml*.

## Spring Security

Spring Security [24] es un *framework* altamente personalizable que proporciona la autenticación, autorización y otras configuraciones de seguridad para las aplicaciones.

Se ha integrado Spring Security al proyecto para proveer seguridad en cuanto a autenticación de usuarios y control de accesos no permitidos.

## Maven

Maven [7] es una herramienta software de gestión y construcción de proyectos en Java. Está basada en el concepto POM (*Project Object Model* o Modelo de Objeto de Proyecto) y se configura a través del formato XML.

Los proyectos de Vaadin son en el fondo proyectos Maven, básicamente. Cuando se añaden las dependencias al proyecto de Vaadin con Spring Boot se hace a través de su archivo *pom.xml*, como se ha comentado anteriormente.

## 4.2. Eclipse

Eclipse [6] es un IDE basado en Java, de código abierto y multiplataforma. También dispone un editor de texto con resaltado de sintaxis, permite la integración de pruebas unitarias con JUnit, permite integración con Ant y tiene asistentes para crear proyectos, clases, tests, etc.

## 4.3. Base de datos

Se ha trabajado con la base de datos de PostgreSQL principalmente por conocimiento previo de la misma a lo largo de la carrera ya que se ha tratado en varias asignaturas, y por su fácil gestión en cuanto a integración en la aplicación y el *hosting elegido*.

Además, con la su herramienta de gestión pgAdmin, se facilita la administración y mantenimiento de la base de datos al poder trabajar con una interfaz gráfica en lugar de tener que trabajar por línea de comandos. También facilita scripts de creación, inserción, borrado... , lo cual agiliza el trabajo con la misma, minimizando el margen de error.

### PostgreSQL

PostgreSQL [11] es un SGBD (Sistema Gestor de Bases de Datos) relacional, objeto-relacional y de código abierto.

### pgAdmin

pgAdmin [17] es una potente herramienta para poder diseñar, administrar y realizar el mantenimiento de bases de datos de PostgreSQL.

## 4.4. Gestión del proyecto y control de versiones

Al igual que ocurría con la base de datos, se ha decidido trabajar con estas herramientas para llevar a cabo la gestión del proyecto y su correspondiente

control de versiones por conocimiento previo y facilidad de trabajo, pues por ejemplo GitHub Desktop simplifica mucho la tarea de subir los *commits* al repositorio y ZenHub ayuda a llevar una planificación/organización del trabajo que está pendiente, en proceso, en revisión o realizado.

## GitHub

GitHub [9] es una de las plataformas de repositorios online colaborativos más conocidas, la cual permite llevar a cabo la gestión de proyectos y el control de versiones.

## ZenHub

ZenHub [27] es una plataforma de gestión de proyectos totalmente integrada en GitHub.

Conecta las *issues* creadas en el repositorio de GitHub, permitiendo organizarlas en el tablero *canvas* según estén recién creadas (*New Issues*), pendientes (*To Do*), en proceso (*In Progress*), etc.

También permite modificar el usuario al que está asignado, las etiquetas, los *milestones*, así como añadir comentarios y cerrar la *issue*.

## GitHub Desktop

La herramienta Github Desktop [10] simplifica la tarea la tarea de conectar el repositorio GitHub para llevar a cabo el control de versiones, sin necesidad de usar la línea de comandos de Git.

## 4.5. Overleaf

Overleaf [16] es el editor online de L<sup>A</sup>T<sub>E</sub>X. Es un editor de textos orientado a crear documentos escritos de alta calidad tipográfica.

Permite llevar un control de versiones, compartir los documentos para colaborar en tiempo real, hacer revisiones, sin necesidad de instalar nada.

## 4.6. *Hosting* para despliegue

- Herramientas consideradas: [LucusHost](#), [000webhost](#), [X10hosting](#), [Awardspace](#), [Runhosting](#), [Batcave](#), [Freehostia](#), [GitHub Pages](#), [Nanobox](#) y

[Heroku](#).

- Herramienta elegida: [Heroku](#).

Finalmente se optó por trabajar con Heroku [18], ya que tras muchas pruebas con el resto de opciones, fue la única con la que se consiguieron resultados positivos.

Heroku es una plataforma de aplicación como servicio (PaaS) completamente administrada y basada en la nube, para poder crear, ejecutar y administrar las aplicaciones web.

Permite importar el código directamente desde GitHub configurando el repositorio, e incluir *add-ons* de las bases de datos como PostgreSQL sin tener que realizar ninguna configuración adicional en la aplicación para conectarla.

## 4.7. Herramientas de diseño

### StarUML

StarUML [13] es una aplicación de escritorio que permite crear diseños y diagramas UML, como diagramas de casos de uso, de clases, de secuencia, etc. desde una interfaz muy sencilla.

### Creately

Creately [1] es otra herramienta de diseño de diagramas online, que permite colaboración en tiempo real (hasta 3 colaboradores de forma gratuita) y diseño a partir de plantillas base.

### Balsamiq Wireframes

Balsamiq Wireframes [1] es una herramienta que permite diseñar el interfaz de las aplicaciones.

Ayuda a realizar una maqueta del software que se va a crear, lo cual resulta muy útil para llevar a cabo un planteamiento inicial del desarrollo de la aplicación.





---

# Aspectos relevantes del desarrollo del proyecto

---

En este apartado se van a detallar los aspectos más importantes del proyecto, como son las decisiones tomadas y los errores y su resolución.

## 5.1. Motivación de la elección del proyecto

El diseño de aplicaciones, más aún de aplicaciones web, siempre algo que me ha llamado la atención, por lo que en cuanto vi la oportunidad de poder desarrollar una por mí misma no me lo pensé.

Además, la idea de poder trabajar con Vaadin para programar en Java una aplicación web me aportó más seguridad, pues es un lenguaje con el que hemos trabajado casi prácticamente los 4 años de carrera.

## 5.2. Periodo de prueba y toma de contacto más largo de lo esperado

Al tratarse de algo prácticamente nuevo, y trabajar con tecnologías como son las APIs de Google, realizar un proyecto de prueba para tomar contacto con las mismas se consideró algo crucial.

La investigación comenzó por crear un proyecto de Vaadin 8, generando la base del mismo desde Eclipse con el *plugin* de Vaadin.

Los primeros problemas llegaron aquí, ya que para poder ejecutar la aplicación había que configurar e iniciar un servidor web a través del archivo

*pom.xml*, y hasta que se consiguió configurar correctamente el servidor HTTP Jetty [8] pasaron varios días.

El siguiente paso fue tratar de incluir las APIs de Google a este proyecto de prueba, y debido a que al principio resultó muy difícil, se optó por crear otro proyecto de prueba de Gradle para después “migrarlo” a Vaadin.

Tras conseguir configurar correctamente la API de Google Calendar en el proyecto de Vaadin 8, se pasó a integrar Google Sign-In siguiendo los diferentes tutoriales de configuración, lo cual resultó imposible. La solución planteada fué migrar de Vaadin 8 a Vaadin 10, con lo que se perdió bastante tiempo, pues muchos de los avances conseguidos con la parte de Calendar se tuvieron que replantear para adaptarlos a la nueva versión.

Tras muchas pruebas, el único resultado medianamente convincente de que se podía integrar el componente de Google Sign-In era integrándolo directamente mediante código HTML con comandos de Vaadin, aunque lo único que se consiguió fue poder lanzar la ventana de petición de la cuenta de Google.

En resumen, hasta finales de febrero los únicos resultados aceptables que se consiguieron fueron con la API de Google Calendar, ya que se pudieron crear y obtener eventos (reservas) y calendarios, mostrando lo obtenido por pantalla mediante la *app* de Vaadin.

### 5.3. Vaadin

Haber elegido Vaadin para desarrollar una aplicación web, en lugar de *lanzarse a la piscina* con lenguajes como HTML, JavaScript, ASP.NET..., conociendo bien el lenguaje Java ha sido una de las mejores opciones.

Con esta tecnología se ha podido desarrollar una aplicación eficiente y con un diseño limpio y ordenado, todo gracias a sus componentes prediseñados (fácilmente modificables estéticamente) y su infinidad de tutoriales y ejemplos disponibles en su web.

### 5.4. Cambio de versión de Vaadin y fracaso en la integración de las APIs de Google

Cuando se empezó a crear el proyecto definitivo, tras el periodo de prueba, se replanteó el uso de la versión 10 de Vaadin, ya que había opciones

más convincentes a partir de la versión 14 que permitían integrar tecnologías como Spring Boot que facilitan el desarrollo.

Además, con Spring Boot parecía que la integración de APIs de Google iba a resultar más sencilla aún, pues había muchas aplicaciones con ellas funcionando.

Avanzado el desarrollo se cambió sucesivas veces de versiones, primero a una versión 15 y después a Vaadin 16, ya que la 15 era simplemente una solución temporal. Lo cual, semanas más tarde, cuando el TFG estuvo ya en su fase final, y tras tener una experiencia más profunda con Vaadin 16, se concluyó que fue una decisión equivocada, ya que la versión estable y de largo mantenimiento es la 14.

Avanzado el desarrollo, y tras haber implementado muchas de las funcionalidades, se volvió a intentar integrar el componente de Google Sign-In, que una vez más no obtuvo resultados exitosos.

En la propia página de Vaadin hay explicaciones de cómo integrar el Sign-In a la aplicación, y tras intentarlo repetidas veces sin éxito se optó por preguntar tanto en los foros como en la página de explicación, sin obtener ninguna respuesta.

Al no conseguir resultados, se pasó a integrar de nuevo la API de Google Calendar, que resultó otro fracaso, ya que en la versión 16 de Vaadin no se consiguió introducir el código desarrollado en el proyecto de pruebas, ni ninguna aproximación al mismo que lo hiciese funcionar.

En conclusión, ninguna de las pruebas realizadas al inicio del proyecto, que funcionaban en Vaadin 8 y 10, se han podido integrar en la aplicación final. Tampoco se podía retroceder a versiones anteriores sin invertir mucho tiempo en ello y renunciar a los nuevos componentes integrados disponibles a partir de la versión de Vaadin 14. Por lo que debido al tiempo que quedaba para la entrega final, se optó por descartar la integración de las tecnologías de Google.

## 5.5. Spring Security

Para compensar la carencia de las APIs de Google, que iban a ser el punto fuerte de la *app* desarrollada, se optó por integrar Spring Security para proveer el login de la aplicación.

Con esta tecnología se consiguieron grandes avances, ya que además de añadir una ventana de login, se han realizado las configuraciones de

seguridad, controlando el acceso de usuarios no autorizados a las ventanas, redirigiéndolos al login, o mostrando ventanas de acceso denegado.

## 5.6. Spring Data JPA

Trabajar con JPA, en concreto con Spring Data JPA, supone una gran ventaja en cuanto a independencia del SGBD, ya que se interactúa directamente con objetos Java que mapean (hacen referencia) a los datos del modelo relacional.

Además, gracias a su integración a través de Spring, no es necesario crear el archivo en el que se definen las relaciones de los objetos con los datos, si no simplemente incluir la dependencia “*Spring Boot Starter Data JPA*” al *pom.xml*.

## 5.7. Uso del interfaz *Specification* de Spring Data JPA para las consultas

Debido a que tanto la consulta de reservas como la consulta de disponibilidad de aulas varía en función a los parámetros que el usuario introduzca, se tomó la decisión de usar las *Specification* que provee Spring Data JPA [2].

Implementando dicha interfaz, se generan consultas individuales para cada parámetro que haya introducido el usuario, y en función de si es nulo o no se añade a la consulta final para obtener los datos.

## 5.8. Consulta de disponibilidad de aulas

Ya que la consulta de la disponibilidad de aulas tenía su *qué* en cuanto a nivel de complejidad, y al ser una de las partes críticas del proyecto, se decidió someterla a pruebas con ayuda de Spring Boot.

De esta forma a medida que se iba refinando la consulta, se iba probando que los cambios realizados producían los resultados esperados, sin tener que estar constantemente realizando las pruebas de forma manual y evitando equivocaciones en la interpretación de dichos resultados.

---

## Trabajos relacionados

---

En este apartado se comentarán aplicaciones y trabajos de carácter similar, relacionados con las reservas.

Una de las ventajas del presente proyecto es que está enfocado a la gestión de reservas de los espacios de la universidad, ya que se trabajan con detalles como la capacidad de las aulas o el número de ordenadores que hay en las mismas.

Otro detalle que cabe destacar es que se permite a cualquier usuario consultar las reservas y la disponibilidad de las aulas, pudiéndose realizar un filtrado por fechas, horas, días, centros, aulas, etc. para obtener búsquedas más concretas y detalladas.

Además, otra de las ventajas es que la reserva de aulas queda restringida al responsable del centro o departamento propietario del aula que se quiere reservar, de forma que no se satura el sistema con peticiones de reserva de cualquier persona para su aprobación.

### **6.1. Sistema Interno de Reserva de Espacios de la EPS de la Universidad de Sevilla**

El propio Laboratorio de Informática de la Escuela Politécnica Superior de la Universidad de Sevilla ha desarrollado una aplicación para gestionar los espacios de la propia escuela [5].

## 6.2. Booked

Booked [21] es un programador de reservas de pago disponible en 40 idiomas, que permite gestionar las reservas de aulas, salas de reuniones, etc., los recursos (aulas y salas que se pueden reservar), y visualizar las reservas en un calendario.

## 6.3. SuperSaaS

SuperSaaS [20] es un planificador de reservas online. Como indican en su web está adaptada a cualquier tipo de negocio permitiendo programación uno a uno (para terapeutas, entrenadores personales, etc), programación de grupo (gimnasios, turismo, escuelas, etc.), reservas y alquileres (alquiler de espacios, planificación de recursos) y reserva de citas (enfocada a salones de belleza, profesionales, peluquerías, etc). Dispone además de una opción gratuita para fines no comerciales.

Una parte interesante de la misma es que permite su integración en el sitio web propio y en Facebook a través de su API y tiene integrado un sistema de pagos.

## 6.4. Acuity Scheduling

Acuity Scheduling [19] es un asistente personal de la agenda enfocado a empresas de fitness que ofertan clases y sesiones privadas, permitiendo reservar citas, consultar la disponibilidad en tiempo real y el pago por adelantado. Tiene disponible también una versión gratuita y ofrece un periodo de prueba gratis.

---

# Conclusiones y Líneas de trabajo futuras

---

## 7.1. Conclusiones

Tras finalizar el proyecto se puede concluir que desarrollar una aplicación web con Vaadin, puede resultar bastante sencillo, sobre todo cuando apenas se tienen conocimientos sobre lenguaje web. Poder trabajar con Java, que es un lenguaje muy extendido, y con tecnologías como Spring Boot que facilitan la configuración de otras dependencias es una de las grandes ventajas aportadas por esta plataforma.

Por el contrario, un punto negativo que tiene esta plataforma es el numeroso lanzamiento de versiones y el mantenimiento a largo plazo de muchas de ellas. Debido a esto, y a que no queda muy claro en su documentación qué versión es ventajosa para ciertas cosas frente a otras, resulta una tarea difícil elegir la versión adecuada desde el principio. Además, en un TFG en el que tampoco se dispone de un largo periodo para el desarrollo, llegado cierto punto es inconcebible hacer el cambio de versión.

La curva de aprendizaje de Vaadin resultó elevada al principio, ya que al integrar otras tecnologías como Spring Boot y tratarse de algo bastante nuevo para mí, saber configurar bien cada cosa no fue sencillo. Pero cabe destacar que en su página disponen de gran cantidad de tutoriales y explicaciones, por lo que superado el pequeño bache del inicio y pese a la difícil tarea de intentar integrar las herramientas de Google, el resto fue rodado.

A pesar de que el Trabajo de *Fin* de Carrera se corresponde a 12 créditos, unas 300 horas de trabajo, un desarrollo web integrando varias herramientas externas conlleva muchísimo más tiempo, y más cuando apenas se ha tratado

este tema. De hecho ha sido difícil compatibilizarlo con la realización de las prácticas curriculares y extracurriculares, pues a partir de febrero sólo disponía de la mitad de tiempo.

El presente trabajo ha servido para aplicar muchos de los conocimientos aprendidos a lo largo de la carrera, y en el periodo de prácticas, en cuanto a trabajo relacionado con la base de datos, desarrollo de código eficiente, intentando evitar la duplicación agrupándolo en clases comunes e implementación de una interfaz gráfica lo más amigable y visualmente agradable posible. Lo cual se puede considerar la finalidad última de un TFG.

## **7.2. Líneas de trabajo futuras**

A pesar de entregar un producto completamente funcional que cumple con el objetivo de la reserva de aulas, se pueden integrar otras funcionalidades al mismo para completarlo:

- Enviar un correo de confirmación tras la realización o modificación de una reserva.
- Cuando las nuevas versiones de Vaadin lo permitan, integrar la API de Google Calendar y el componente de Google Sign-In para poder insertar los eventos en el calendario personal del usuario responsable.
- Mostrar el calendario con las reservas en una interfaz modo agenda, en la ventana de Consulta de Reservas para que resulte más visual.
- Incluir la internacionalización de la aplicación.



---

## Bibliografía

---

- [1] Cinergix Pty Ltd (Australia). Creately. <https://creately.com/es/home/>. [Internet; Accedido 15-julio-2020].
- [2] baeldung. Rest query language with spring data jpa specifications. <https://www.baeldung.com/rest-api-search-language-spring-data-specifications>. [Internet; Accedido 19-julio-2020].
- [3] Raquel Castaño. ¿qué es un desarrollador full stack? <https://www.neoland.es/blog/que-es-desarrollador-full-stack>, 2018. [Internet; Accedido 15-julio-2020].
- [4] Nicole Chapaval. Qué es frontend y backend. <https://platzi.com/blog/que-es-frontend-y-backend/>, 2018. [Internet; Accedido 15-julio-2020].
- [5] Universidad de Sevilla. Sistema interno de reserva de espacios eps universidad de sevilla. <https://eps.us.es/instalaciones-y-servicios/aulas/reserva-de-aulas>. [Internet; Accedido 18-julio-2020].
- [6] Inc. Eclipse Foundation. Eclipse. [Internet; Accedido 15-julio-2020].
- [7] The Apache Software Foundation. Maven. <https://maven.apache.org/>. [Internet; Accedido 15-julio-2020].
- [8] The Eclipse Foundation. Jetty. <https://www.eclipse.org/jetty/>. [Internet; Accedido 17-julio-2020].
- [9] Inc. GitHub. Github. <https://github.com/>. [Internet; Accedido 15-julio-2020].

- [10] Inc. GitHub. Github desktop. <https://desktop.github.com/>. [Internet; Accedido 15-julio-2020].
- [11] The PostgreSQL Global Development Group. Postgresql. <https://www.postgresql.org/>. [Internet; Accedido 15-julio-2020].
- [12] JavaScript.com. Javascript. <https://www.javascript.com/>. [Internet; Accedido 15-julio-2020].
- [13] Ltd. MKLabs Co. Staruml. <http://staruml.io/>. [Internet; Accedido 15-julio-2020].
- [14] Oracle. Java. <https://www.java.com/es/>. [Internet; Accedido 15-julio-2020].
- [15] Angel Eulises Ortiz. Qué es un framework en informática o programación. [https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/#:~:text=La%20palabra%20Framework%20es%20la,%20y%20Trabajo%20\(work\)%20.&text=En%20otras%20palabras%2C%20podemos%20decir,estandarizadas%20para%20crear%20aplicaciones%20web.,](https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/#:~:text=La%20palabra%20Framework%20es%20la,%20y%20Trabajo%20(work)%20.&text=En%20otras%20palabras%2C%20podemos%20decir,estandarizadas%20para%20crear%20aplicaciones%20web.,) 2018. [Internet; Accedido 15-julio-2020].
- [16] Overleaf. Overleaf. <https://es.overleaf.com/>. [Internet; Accedido 15-julio-2020].
- [17] pgAdmin. Postgresql. <https://www.pgadmin.org/>. [Internet; Accedido 15-julio-2020].
- [18] Salesforce.com. Heroku. <https://www.heroku.com/home>. [Internet; Accedido 15-julio-2020].
- [19] Squarespace. Acuity scheduling. <https://acuityscheduling.com/?v3=b>. [Internet; Accedido 18-julio-2020].
- [20] SuperSaaS. Supersaas. <https://www.supersaas.es/>. [Internet; Accedido 18-julio-2020].
- [21] LLC. Twinkle Toes Software. Booked. <https://www.bookedscheduler.com/>. [Internet; Accedido 18-julio-2020].
- [22] Vaadin. Vaadin. <https://vaadin.com/>. [Internet; Accedido 15-julio-2020].
- [23] Inc. VMware. Spring boot. <https://spring.io/projects/spring-boot>. [Internet; Accedido 15-julio-2020].

- [24] Inc. VMware. Spring security. <https://spring.io/projects/spring-security>. [Internet; Accedido 15-julio-2020].
- [25] W3Schools. Css. <https://www.w3schools.com/css/>. [Internet; Accedido 15-julio-2020].
- [26] Wikipedia. Convención sobre configuración. [https://es.wikipedia.org/wiki/Convenci%C3%B3n\\_sobre\\_configuraci%C3%B3n#:~:text=Convenci%C3%B3n%20sobre%20Configuraci%C3%B3n%2C%20tambi%C3%A9n%20conocido,no%20perdiendo%20flexibilidad%20por%20ello.](https://es.wikipedia.org/wiki/Convenci%C3%B3n_sobre_configuraci%C3%B3n#:~:text=Convenci%C3%B3n%20sobre%20Configuraci%C3%B3n%2C%20tambi%C3%A9n%20conocido,no%20perdiendo%20flexibilidad%20por%20ello.), 2020. [Internet; descargado 15-julio-2020].
- [27] ZenHub. Zenhub. <https://www.zenhub.com/>. [Internet; Accedido 15-julio-2020].