



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**GII 22.24 Aplicación de
gestión de TFGs**



Presentado por David Renedo Gil
en Universidad de Burgos — 16 de enero
de 2023

Tutores: Álvaro Arnaiz González y Ana Serrano
Mamolar



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Álvar Arnaiz González y Dña. Ana Serrano Mamolar, profesores del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. David Renedo Gil, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «GII 22.24 Aplicación de gestión de TFGs».

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, 16 de enero de 2023

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Dña. Ana Serrano Mamolar

D. Álvar Arnaiz González

Resumen

El proyecto propone evolucionar la aplicación web de Gestor de trabajos de fin de grado, una aplicación para el manejo de procesos de oferta, búsqueda, asignación y evaluación de los TFG de la carrera de Ingeniería Informática en la Universidad de Burgos. Usando como lenguaje de programación Java y empleando Vaadin como framework para los componentes gráficos, se requieren mejoras visuales, así como la creación de nuevas pantallas. Adicionalmente se ha implementado un sistema de extracción de información mediante *websrcaping* para crear informes y presentar datos, además de dar la posibilidad de asignar y ofertar TFGs directamente desde la aplicación. También se da la oportunidad de cambiar la información de los TFGs activos a los usuarios administradores.

Descriptores

Aplicación web, Vaadin, Eclipse IDE, Java, trabajo de fin de grado, gestor de proyectos, Moodle, Maven, Apache Tomcat, WebScraping.

Abstract

The project proposes to evolve the fina degree prject manager web application, in charge of managing the offer, search, assignment and evaluation of Computer Engineering career final projects career of the University of Burgos. It uses Java as programming language and employs graphic components of the Vaadin framework. Visual improvements are required, as well as the creation of new screens. Additionally, an information extraction system has been implemented using webscraping to create reports and present data, and the possibility of assigning and bidding TFGs directly from the application. It is also given the opportunity to change the information of the active projects by the administrator users.

Keywords

Web Application, Vaadin, Eclipse IDE, Java, final degree project, project manager, Moodle, Maven, Apache Tomcat, WebScraping.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
1.1. Estructura de la memoria	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
Conceptos teóricos	5
3.1. <i>Webscraping</i>	5
3.2. <i>Desarrollo Backend</i>	5
3.3. <i>Desarrollo Frontend</i>	5
3.4. <i>Framework</i>	6
3.5. <i>ApexCharts</i>	7
3.6. <i>Moodle API</i>	7
Técnicas y herramientas	9
4.1. Herramientas de Desarrollo	9
4.2. Herramientas de Documentación	11
Aspectos relevantes del desarrollo del proyecto	13
5.1. Problemas a la hora de ejecutar el proyecto	13
5.2. WebScraping	16

5.3. Trabajo con Vaadin	16
5.4. Actualización de ficheros	17
Trabajos relacionados	19
6.1. GII 20.09 Herramienta web repositorios de TFGII	19
6.2. Guardians. Development of a web application facilitating communication between teachers and guardians.	20
6.3. Aplicación web para la gestión de TFG's en una universidad	20
Conclusiones y Líneas de trabajo futuras	21
7.1. Conclusiones	21
7.2. Líneas de trabajo futuras	22
Bibliografía	25

Índice de figuras

5.1. Error tras desplegar el .war en el Gestor de Aplicaciones de Tomcat	14
5.2. Logs proporcionados por Tomcat	14
5.3. Logs proporcionados por Tomcat	15
5.4. Cambio de versión Dynamic Web Module	15

Índice de tablas

Introducción

El proyecto se centrará en la mejora de la actual aplicación del Gestor de asignaciones de TFG, utilizada en el grado de Ingeniería Informática. La aplicación actual se trata de una actualización previa del TFG denominado **GII 20.09 Herramienta web repositorios de TFGII**.

1.1. Estructura de la memoria

La memoria consta de los siguientes apartados:

- **Introducción:** presentación del proyecto y estructuración de este.
- **Objetivos del proyecto:** exposición de los objetivos generales, técnicos y personales del proyecto.
- **Conceptos teóricos:** explicación de los términos teóricos necesarios para la comprensión y el desarrollo del proyecto.
- **Técnicas y herramientas:** definición de las técnicas utilizadas para el desarrollo del proyecto y de las herramientas empleadas para su funcionamiento.
- **Aspectos relevantes del desarrollo:** breve explicación de los términos más importantes durante el desarrollo del proyecto.
- **Trabajos relacionados:** descripción de los trabajos y proyectos asociados con la gestión de trabajos de fin de grado (TFG).
- **Conclusiones y líneas de trabajo futuras:** resolución obtenida al concluir el proyecto y descripción de posibles futuras líneas de trabajo o mejoras.

- **Anexo:** se realiza de forma complementaria un documento de Anexo con explicación más detallada de todo el desarrollo del proyecto.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto.

2.1. Objetivos generales

- Corregir *bugs* de la versión anterior.
- Crear nuevas pantallas para completar la aplicación.
 - Pantalla con la visualización del histórico de los profesores/áreas/departamentos.
 - Permitir crear un informe de representación de datos de profesores y áreas.
 - Implementar la opción de oferta/asignación de TFGs.
 - Implementar la funcionalidad de aceptar/denegar TFGs propuestos, así como de modificar los proyectos de la pestaña de activos.

2.2. Objetivos técnicos

- Implementar *webscraping* con la página **de investigadores de la UBU**.
- Permitir la oferta de proyectos desde la propia aplicación siendo profesor y la aceptación de los mismos por parte de un administrador.
- Crear una nueva gráfica para las estadísticas de los profesores con las opciones marcadas por el usuario.

- Guardar datos generados en archivos *csv* y *xls*.
- Actualizar los ficheros de datos del profesorado.
- Utilizar GitHub para llevar a cabo el seguimiento del proyecto y control de versiones.
- Registrar el usuario que inicia sesión en el sistema para asignarle un rol y mostrarle las pantallas correspondientes a dicho actor.

Conceptos teóricos

En este apartado se definirán algunos de los conceptos utilizados a lo largo del proyecto.

3.1. *Webscraping*

El *Webscraping* [11] es una técnica que se utiliza para la extracción y almacenamiento de información de cualquier página web a través de un programa de software, que suele ser un *crawler*. Aunque, a priori, puede parecer que cualquiera puede scrapear información en cualquier sitio web, lo cierto es que la extracción de datos no siempre es legal. Por ejemplo, los datos que requieren un registro del usuario no pueden ser obtenidos a través del web scraping.

3.2. *Desarrollo Backend*

Dentro del desarrollo web, el *backend* [15] se encarga de todos los procesos necesarios para que la web funcione de forma correcta. Estos procesos o funciones no son visibles, pero tienen mucha importancia en el buen funcionamiento de un sitio web. Algunas de estas acciones que controla el backend son la conexión con la base de datos o la comunicación con el servidor de hosting.

3.3. *Desarrollo Frontend*

Frontend [16] es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que

vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD: Responsive Web Design), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

3.4. *Framework*

Los **framework** [9] web son un conjunto de herramientas, estilos y librerías dispuestas a través de una estructura o esqueleto base, para el desarrollo de aplicaciones web más escalables y sencillas de mantener. Gracias a estos frameworks web, podemos ahorrar grandes cantidades de tiempo y costes, pero vamos a profundizar más en las ventajas que tienen, causantes de su gran éxito y expansión.

- Documentación y comunidad. La cantidad de documentación que podremos encontrar sobre un framework web, suele ser enorme y además con una gran comunidad detrás, respondiendo preguntas y desarrollando nuevas funcionalidades.
- Reutilización del código. Uno de los puntos fuertes de los frameworks web es la modularidad de su código y la capacidad para poder hacer múltiples proyectos con el mismo código, cambiando simplemente los textos.
- Arquitectura y metodología. La mayoría de frameworks web del mercado usan arquitecturas y metodologías actuales, como el Modelo-Vista-Controlador.
- Plantillas web. Las plantillas facilitan mucho el trabajo de los desarrolladores web y los frameworks no se quedan atrás en esto. Algunos frameworks Frontend como **Bootstrap** cuentan con grandes cantidades de plantillas y componentes desarrollados por su extensa comunidad.
- Seguridad web. Los frameworks web suelen contar con medidas de seguridad para proteger nuestros datos y los de nuestros clientes, ayudando en gran medida en uno de los temas que lleva de cabeza a grandes empresas de servicios web desde el 2017.
- Posicionamiento en motores de búsqueda. El **posicionamiento web SEO** on page es muy importante si queremos lograr aparecer en las primeras posiciones de buscadores como Google. Por eso muchos frameworks

web ya implementan en su estructura código para poder lograrlo más fácilmente.

- Ofertas de empleo. Si una persona llega a aprender a usar un framework web de manera solvente, esa persona tendrá muchas posibilidades de encontrar un trabajo más fácilmente y con un sueldo mayor.

3.5. *ApexCharts*

Apexcharts [2] es una biblioteca de gráficos moderna que ayuda a los desarrolladores a crear visualizaciones atractivas e interactivas para páginas web. Es un proyecto de código abierto con licencia del MIT y es de uso gratuito en aplicaciones comerciales

3.6. *Moodle API*

Moodlelib API [14] es el archivo de la biblioteca central de diversas funciones de Moodle de uso general. Las funciones pueden superar el manejo de parámetros de solicitud, configuraciones, preferencias del usuario, tiempo, inicio de sesión, mnet, complementos, cadenas y otros. También hay muchas constantes definidas.

Moodle [13] es una plataforma de aprendizaje diseñada para proporcionarle a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados. Moodle es proporcionado gratuitamente como programa de **Código Abierto**, bajo la **Licencia Pública General GNU (GNU General Public License)**. Cualquier persona puede adaptar, extender o Modificar Moodle, tanto para proyectos comerciales como no-comerciales, sin pago de cuotas por licenciamiento, y beneficiarse del costo/beneficio, flexibilidad y otras ventajas de usar Moodle. La implementación de Moodle en código abierto significa que Moodle es continuamente revisado y mejorado, para adecuarse a las necesidades actuales y cambiantes de sus usuarios.

Técnicas y herramientas

En esta sección se detallarán algunas de las técnicas y herramientas utilizadas durante el desarrollo del trabajo y las alternativas que no se han llegado a usar.

Muchas de las herramientas utilizadas están ya explicadas en la anterior versión del proyecto que podemos encontrar en [GII 20.09 Herramienta web repositorios de TFGII](#). Por lo tanto lenguajes de programación como *Java*, *SQL*, *XML* no se detallarán a pesar de haber sido utilizados, y herramientas como *GitHub*, *GitHub Desktop*, *Eclipse IDE*, *Tomcat* o *Maven* tampoco se explicarán, ya que además de ser muy comunes, las encontramos explicadas previamente.

4.1. Herramientas de Desarrollo

Heroku para estudiantes

Heroku [6] es una plataforma basada en la nube como servicio (PaaS) para construir, ejecutar y administrar aplicaciones. La plataforma Integración de GitHub le permite conectar su aplicación Heroku a su repositorio GitHub e implementar en cada empuje a GitHub. Heroku ofrece una gama de servicios de bajo costo para ayudarlo a experimentar, aprender y crear prototipos de nuevas ideas. Para los estudiantes de GitHub, vamos un paso más allá y agregamos aún más recursos a su Paquete de desarrolladores de estudiantes de GitHub. Heroku pasó a ser de pago el 28 de noviembre de 2022, pero al pertenecer al programa de GitHub para estudiantes proporcionado por la UBU, podemos solicitarlo, y que se nos agreguen unos créditos para poder desplegar nuestro proyecto en la nube. Por lo que es la herramienta que hemos utilizado para desplegar nuestro proyecto.

Webscrap

Para el desarrollo del *webscraping* se han estudiado diferentes alternativas, entre ellas cabe mencionar:

- Jsoup [10] es una biblioteca Java para trabajar con HTML del mundo real. Proporciona una API muy conveniente para recuperar URL y extraer y manipular datos, utilizando los mejores métodos HTML5 DOM y selectores CSS. Es la herramienta que hemos utilizado para realizar el *webscraping* ya que es la que mejor se ajustaba y con la que mejor resultado hemos obtenido, también, tras haber intentado utilizar otras alternativas resulta la más sencilla de utilizar.
- HtmlUnit [7] es un «navegador sin GUI para programas Java». Modela documentos HTML y proporciona una API que le permite invocar páginas, completar formularios, hacer clic en enlaces, etc... tal como se hace en un navegador. Es una de las alternativas planteadas a la hora de utilizar una biblioteca para el *webscraping*.

Este recurso es muy utilizado para obtener información de *páginas webs* externas, sin embargo, hay que tener cuidado con las páginas de las que se obtienen los datos. Por ejemplo, los datos que requieren un registro del usuario no pueden ser obtenidos a través del *web scraping*. Tampoco es legal el uso de esta técnica de rastreo para ocultar publicidad, descargas de responsabilidad o términos y condiciones. [11]

Pencil

Pencil [4] está construido con el propósito de proporcionar una herramienta de creación de prototipos GUI gratuita y de código abierto que las personas pueden instalar y usar fácilmente para crear maquetas en plataformas de escritorio populares. Se ha utilizado para crear los *mock-ups* de las nuevas pantallas que se han desarrollado.

Northflank

Northflank [3] es una plataforma de servicio en la nube (Paas) como Heroku. Es un sitio web que cabe mencionar ya que los precios escalan junto a como va desarrollándose tu proyecto. Puedes pagar solo por los recursos que consumen tus servicios durante la compilación y el tiempo de ejecución. Puedes escalar tanto horizontal como verticalmente sin gastos inesperados.

Al ser un VPS, puedes trabajar con muchos lenguajes y frameworks. Cuentan con una versión para Developers que es gratuita. Finalmente se optó por no utilizar este sistema ya que tras intentar importar nuestro proyecto nos indicaba que era de pago por que superaba el límite de peso permitido para un proyecto gratuito.

Vaadin

Vaadin [5] es un framework de desarrollo de SPA que permite escribir el código de dichas aplicaciones en Java o en cualquier otro lenguaje soportado por la JVM 1.6+. Esto permite la programación de la interfaz gráfica en lenguajes como Java 8, Scala o Groovy, por ejemplo.

Uno de las características diferenciadores de Vaadin es que, contrario a las librerías y frameworks de JavaScript típicas, presenta una arquitectura centrada en el servidor, lo que implica que la mayoría de la lógica es ejecutada en los servidores remotos. Del lado del cliente, Vaadin está construido encima de Google Web Toolkit, con el que puede extenderse.

LucidChart

LucidChart [12] es una aplicación de diagramación inteligente que reúne a los equipos para que tomen mejores decisiones y construyan el futuro. Ha sido utilizada para realizar los diagramas de flujo de los anexos.

4.2. Herramientas de Documentación

Latex

Latex [18] es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.

TexStudio

TexStudio [1] es un entorno de escritura integrado para crear documentos LaTeX. Su principal objetivo es hacer que escribir L^AT_EX sea lo más fácil y cómodo posible. Por esto, TeXstudio nos ofrece a los usuarios numerosas funciones como el resaltado de sintaxis, un visor integrado, la verificación

de referencias y varios asistentes, entre otras. Es el programa de edición que hemos utilizado para realizar la memoria y los anexos.

MiKTeX

MiKTeX [1] es una implementación de TeX y programas relacionados para Windows. TeX es un sistema de caracteres de alta calidad para la edición de documentos. Es la herramienta que se utiliza para compilar los documentos que se crean en TexStudio.

Overleaf

Overleaf [17] es una herramienta de publicación y redacción colaborativa en línea que hace que todo el proceso de redacción, edición y publicación de documentos científicos sea mucho más rápido y sencillo. Overleaf ofrece la conveniencia de un editor LaTeX fácil de usar con colaboración en tiempo real y la salida totalmente compilada producida automáticamente en segundo plano a medida que escribe.

Era la herramienta que se iba a utilizar en un principio pero se deshechó la idea ya que se vió que era más intuitivo de utilizar la aplicación de escritorio TexStudio.

Aspectos relevantes del desarrollo del proyecto

Este apartado recoge los aspectos más interesantes del desarrollo del proyecto.

5.1. Problemas a la hora de ejecutar el proyecto

A la hora de ejecutar el proyecto anterior surgieron una serie de problemas tanto para la ejecución por terminal como desde Eclipse.

Cuando quise ejecutarlo mediante la terminal desplegando el archivo `.war` generado tras compilar me surgía el siguiente error, ver imagen 5.1.

Tras buscar información sobre el posible error, se descubre en los logs que proporciona tomcat lo siguiente, ver imagen 5.2. En el que se informa que se intenta ejecutar un proyecto con un tamaño mayor al que tenemos configurado en tomcat.

Para solucionar este problema se accede al archivo *apache-tomcat-9.0.68-webapps-manager-WEB-INF* y se modifican las siguientes líneas, ver imagen 5.3 aumentando el número que se indica.

Cuando quise ejecutarlo mediante Eclipse no me dejaba añadir el proyecto al servidor de *tomcat*, indicando que las versiones no eran compatibles. Por ello se ha entrado en las propiedades del proyecto y se ha cambiado la versión del parametro *Dynamic Web Module* a la 3.1 en el apartado *Project Facets* como se aprecia en la imagen 5.4.

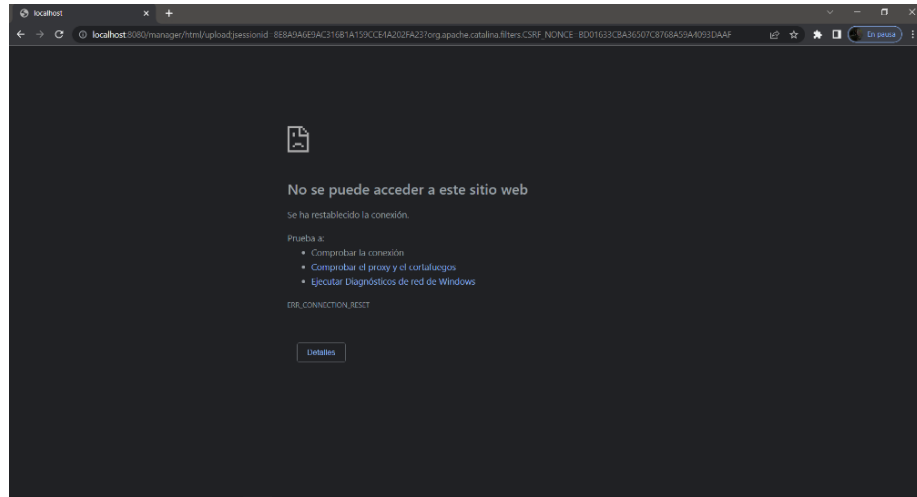


Figura 5.1: Error tras desplegar el .war en el Gestor de Aplicaciones de Tomcat

```

08-Nov-2022 19:17:41.644 INFO [http-nio-8080-exec-2] org.apache.catalina.core.ApplicationContext.log HTMLManager: Init: Associated with Deployer 'Catalina:type=Deployer,host=localhost'
08-Nov-2022 19:17:41.644 INFO [http-nio-8080-exec-2] org.apache.catalina.core.ApplicationContext.log HTMLManager: Init: Global resources are available
08-Nov-2022 19:17:41.655 INFO [http-nio-8080-exec-2] org.apache.catalina.core.ApplicationContext.log HTMLManager: list: Listing contexts for virtual host 'localhost'
08-Nov-2022 19:18:02.406 SEVERE [http-nio-8080-exec-4] org.apache.catalina.core.ApplicationContext.log HTMLManager: FailD - Fallo carga de despliegue, Excepción: [org.apache.tomcat.util.http.fileupload.impl.SizeLimitExceededException: org.apache.tomcat.util.http.fileupload.impl.SizeLimitExceededException: the request was rejected because its size (72771942) exceeds the configured maximum (52428800)]
at org.apache.catalina.connector.Request.parseParameters(Request.java:3275)
at org.apache.catalina.connector.RequestFacade.getParameter(Request.java:1241)
at org.apache.catalina.connector.RequestFacade.getParameter(RequestFacade.java:381)
at org.apache.catalina.filters.CsrfPreventionFilter.doFilter(CsrfPreventionFilter.java:127)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
at org.apache.catalina.filters.HttpHeaderSecurityFilter.doFilter(HttpHeaderSecurityFilter.java:126)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:157)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:169)
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:680)
at org.apache.catalina.valves.RequestFilterValve.process(RequestFilterValve.java:376)
at org.apache.catalina.valves.RemoteAddrValve.invoke(RemoteAddrValve.java:16)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:135)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:82)
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:687)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:78)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:360)
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:399)
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:65)
at org.apache.coyote.AbstractProtocolConnectionHandler.process(AbstractProtocol.java:893)
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1789)
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1193)
at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.base/java.lang.Thread.run(Thread.java:854)
Caused by: org.apache.tomcat.util.http.fileupload.impl.SizeLimitExceededException: the request was rejected because its size (72771942) exceeds the configured maximum (52428800)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:183)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:205)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:224)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:142)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:252)
at org.apache.tomcat.util.http.fileupload.impl.FileItemIteratorImpl.getItemIteratorImpl.java:276)
at org.apache.catalina.connector.Request.parseParameters(Request.java:3275)
... 51 more
08-Nov-2022 19:18:02.406 INFO [http-nio-8080-exec-4] org.apache.catalina.core.ApplicationContext.log HTMLManager: list: Listing contexts for virtual host 'localhost'

```

Figura 5.2: Logs proporcionados por Tomcat

5.1. PROBLEMAS A LA HORA DE EJECUTAR EL PROYECTO 15



```
<?xml version='1.0' encoding='UTF-8' request-character-encoding='>

<servlet>
  <servlet-name>Manager</servlet-name>
  <servlet-class>org.apache.catalina.manager.ManagersServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>2</param-value>
  </init-param>
</servlet>
<servlet>
  <servlet-name>HTMLManager</servlet-name>
  <servlet-class>org.apache.catalina.manager.HTMLManagersServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>2</param-value>
  </init-param>
  <!-- Uncomment this to show proxy sessions from the Backup manager or a
  StoreManager in the sessions list for an application
  <init-param>
    <param-name>showProxySessions</param-name>
    <param-value>true</param-value>
  </init-param>
  -->
  <multipart-config>
    <!-- GMP max -->
    <max-file-size>524288000</max-file-size>
    <max-request-size>524288000</max-request-size>
    <file-size-threshold>0</file-size-threshold>
  </multipart-config>
</servlet>
<servlet>
  <servlet-name>Status</servlet-name>
  <servlet-class>org.apache.catalina.manager.StatusManagerServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>8</param-value>
  </init-param>
</servlet>
<servlet>
  <servlet-name>Proxy</servlet-name>
```

Figura 5.3: Logs proporcionados por Tomcat

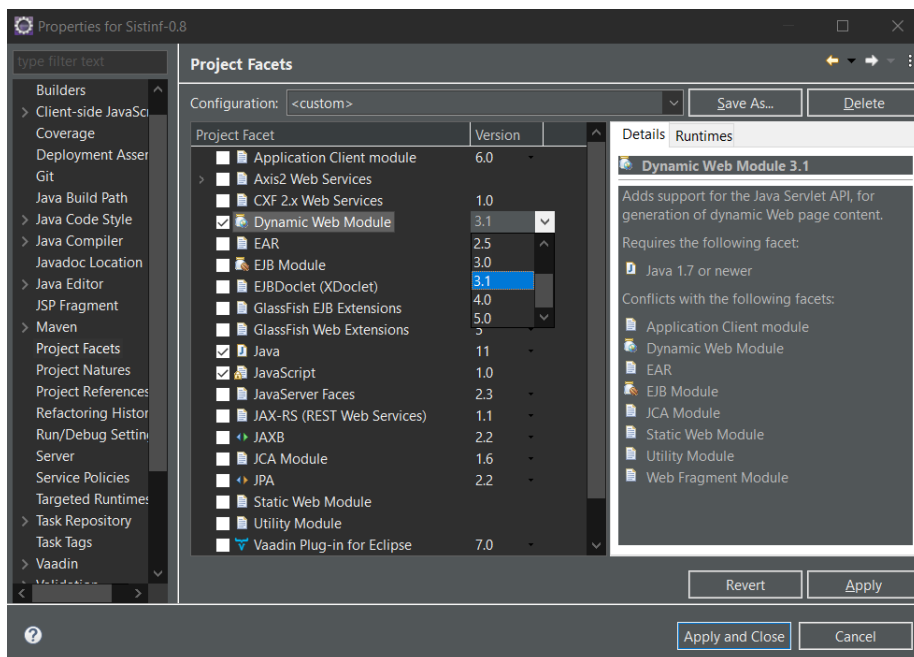


Figura 5.4: Cambio de versión Dynamic Web Module

5.2. WebScraping

Uno de los objetivos del proyecto que teníamos era la obtención de datos sobre los profesores que se encuentran en la EPS mediante *webscraping*, realizando finalmente con la librería **JSoup**, tras debatir su uso con otras librerías planteadas. Con ella hemos sacado los datos: nombre, apellidos, área y departamento de cada uno de los profesores. Una vez obtenemos esta información la utilizamos para actualizar a la última versión las bases de datos que utilizamos durante todo el proyecto. También se ha utilizado la librería **CSVWriter** y la *API*: **WorkbookFactory** para escribir los datos en los ficheros correspondientes.

5.3. Trabajo con Vaadin

Este proyecto se ha tenido que llevar a cabo con *Vaadin*, es la primera vez que se utiliza este *framework*, por lo que se ha tenido que estudiar todo su funcionamiento, y como implantarlo durante el desarrollo.

Se han tenido que utilizar muchos componentes nuevos que no estaban en la versión anterior como pueden ser:

- *TextArea*: componente para guardar texto introducido por teclado por parte del usuario.
- *Checkbox*: componente que permite realizar una selección múltiple de elementos.
- *Binder*: componente para conectar la clase del formulario con la información requerida.
- *ComboBox*: componente desplegable con diferentes datos.
- *Anchor*: componente utilizado para realizar el documento descargable.
- *Dialog*: componente estilo *pop-up* para realizar confirmaciones.

Toda la información sobre el uso de *Vaadin* <https://vaadin.com/> se ha obtenido de su página atendiendo a la versión que se utiliza.

5.4. Actualización de ficheros

Se ha tenido que incorporar en varias ocasiones la creación de archivos *.xls* o *.csv*, así como la actualización en tiempo real de la información de la base de datos, ya sea al introducir un TFG, al modificar su estado, o cuando queremos actualizar la información del *web scraping* sobre los tutores.

Para ello se ha empleado las librerías *CSVWriter* o *WorkBook*, estudiando su funcionamiento.

Trabajos relacionados

Se nombrarán algunos de los proyectos y aplicaciones similares o relacionados con la gestión de trabajos de fin de grado o master que se han visitado para la obtención de información del proyecto.

Varios de los TFGs visualizados son los mismos que menciona Diana en la [versión anterior](#), de la que se hablará, por lo que no se comentarán.

6.1. GII 20.09 Herramienta web repositorios de TFGII

Este el proyecto que se propone para mejorar consiste en una aplicación web para la gestión de TFG de la Universidad de Burgos.

Este proyecto es una actualización de la versión previa encontrada en [repositorio de GitHub](#).

El código fuente se puede encontrar en este [repositorio de GitHub](#). En el anexo se ha explicado como lo hemos compilado y ejecutado para ver su funcionamiento y seguir con el desarrollo. Es un trabajo realizado mediante Java 11, con Vaadin 14 como plataforma de código abierto para la interfaz web.

6.2. Guardians. Development of a web application facilitating communication between teachers and guardians.

Este proyecto es una aplicación realizada mediante *Vaadin* que podemos observar en el siguiente [enlace](#).

Es una aplicación que ayuda y simplifica el trabajo diario de los profesores. El profesor puede crear eventos, examinar a sus alumnos, comprobar el progreso de sus alumnos y otras funcionalidades útiles. Las tecnologías que se han utilizado mayoritariamente son *Spring MVC* y *Vaadin*.

En este proyecto podemos observar el funcionamiento de los roles de los usuarios de una plataforma así como del propio uso de *Vaadin*.

6.3. Aplicación web para la gestión de TFG's en una universidad

Este proyecto tiene como finalidad gestionar los proyectos de una universidad como el nuestro. Podemos ver su funcionamiento en el siguiente [enlace](#).

De este proyecto podemos sacar como está estructurada la aplicación mediante las vistas, y como se organiza mediante DAOS (Organización Autónoma Descentralizadas). Está desarrollado en java y funciona mediante *servlets*. Los *servlets* son programas JavaTM que utilizan la interfaz de programación de aplicaciones (API) de *servlet* Java. Se ejecutan en un servidor web habilitado para Java, amplían las funciones de un servidor web, de forma similar a como se ejecutan *applets* en un navegador, y amplían las funciones de un navegador. [8].

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Una vez finalizado el proyecto y remontando la vista atrás al comienzo del mismo puedo afirmar que se han obtenido muchos conocimientos.

La mayoría de ellos están relacionados con el *framework de Vaadin* y su funcionamiento. El proyecto ha cumplido con todas las propuestas planteadas al inicio del curso y además se han ido supervisando las diferentes formas de adaptarlas mediante ideas de los tutores y mías propias. Por ejemplo, en un primer momento sólo se planteaba la opción de proponer proyectos y posteriormente darle a un administrador la oportunidad de aceptarlos y denegarlos. Pero en las alturas finales del mismo también se decidió darle otra vuelta y permitir al administrador modificar los proyectos que estuviesen activos que quisiese, de esta forma se ha tenido que añadir una funcionalidad extra al proceso.

El proyecto propone mejorar una aplicación ya existente por lo que en gran parte del comienzo implica estudiar y comprender cómo funciona y cómo está estructurada la versión anterior. Gran parte del estudio se ve facilitada gracias a los comentarios que estaban expuestos en el código y que también se han aumentado en esta versión, como ayuda para posteriores mejoras. Se empleó mucho tiempo en conocer el código y sobretodo en entender las herramientas de uso, entre ellas destacar el uso y funcionamiento del ya mencionado *Vaadin* y del la propia herramienta de documentación \LaTeX que nunca había sido utilizada.

Gran parte del funcionamiento se basó en la extracción de información sobre el profesorado de la Escuela Politécnica Superior mediante *webscraping*, cuyos datos se utilizan en la mayoría de las pantallas añadidas. Se estudio como realizar esta técnica, y se intentó mediante diferentes librerías dando finalmente con la solución.

He aprendido mucho sobre el funcionamiento de la aplicación y de los protocolos utilizados para realizar las peticiones entre cliente y servidor. En cuanto a los lenguajes, he reforzado los conocimientos que tenía sobre SQL y sobretodo Java, que se cursan a lo largo del grado. También he aprendido un poco sobre CSS, del cual tenía unos conocimientos muy vagos.

Otro de los aspectos más relevantes que se ha tenido que tratar ha sido la gestión del tipo de usuario que iniciaba sesión en la aplicación, cómo obtener su rol y de esta forma restringir las vistas a las que puede acceder. Se estudiaron numerosas alternativas para lograr esta funcionalidad, como la que se puede observar en al rama *TestBranch* del proyecto y finalmente se implantó la más sencilla y efectiva.

7.2. Líneas de trabajo futuras

Algunas de las posibles líneas de trabajo futuras del programa son:

- Incluir la **internacionalización** de la aplicación, traducirlo a otros idiomas como inglés.
- Añadir una funcionalidad que *avisase a los tutores de un TFG sobre su estado*. Es decir, enviar un mensaje al tutor de un proyecto que haya sido modificado por un administrador, ya sea modificando su estado entre aceptado y denegado o cambiando su información.
- Como se menciona en la anterior versión y que no ha sido implementado se podría añadir la **distinción de sesiones empleando la información de las Cookies**. Esto permitirá el inicio de sesión de varios usuarios a la vez. Con los datos obtenidos de las Cookies se podría obtener la región desde la cual se está accediendo y aplicarlo para calcular la fecha de la última modificación de los ficheros, según la zona horaria.
- Mejorar los aspecto visuales de las nuevas pantallas añadidas, ya que en esta versión se ha tenido más en cuenta la funcionalidad.

- Permitir a un alumno solicitar un proyecto de la lista de activos que existan desde la propia aplicación, que no lo tenga que hacer mediante un correo referenciando dicho proyecto. Es decir añadir una columna en el apartado de activos que pueda ser «Solicitado por».

Bibliografía

- [1] Damián A. Textstudio, un entorno de escritura para crear documentos latex. <https://ubunlog.com/textstudio-crear-documentos-latex/>, 2018. [Internet; Accedido 7-octubre-2022].
- [2] APEXCHARTS.JS. Apexcharts.js. <https://apexcharts.com/>. [Internet; Accedido 10-noviembre-2022].
- [3] Nube Colectiva. Heroku ya no es gratis, aquí estan las 5 mejores alternativas. https://blog.nubecollectiva.com/heroku-ya-no-es-gratis-aqui-estan-las-5-mejores-alternativas/#Las_5_Mejores_Alternativas_a_Heroku, 2022. [Internet; Accedido 12-noviembre-2022].
- [4] Evolus. Pencil project. <https://pencil.evolus.vn/>, 2019. [Internet; Accedido 25-noviembre-2022].
- [5] Ari Handler Gamboa. Introducción a vaadin. <https://www.adictosaltrabajo.com/2015/11/30/introduccion-a-vaadin/>, 2015. [Internet; Accedido 5-octubre-2022].
- [6] Heroku. Heroku para estudiantes de github. <https://www.heroku.com/github-students>, 2022. [Internet; Accedido 16-noviembre-2022].
- [7] HTMLUnit. Htmlunit. <https://htmlunit.sourceforge.io/>. [Internet; Accedido 16-noviembre-2022].
- [8] IBM. Servlets. <https://www.ibm.com/docs/es/was-nd/9.0.5?topic=applications-servlets>. [Internet; Accedido 16-enero-2023].

- [9] illusionstudio. ¿qué es un framework web y qué ventajas aportan? <https://www.illusionstudio.es/que-es-un-framework-web>, 2018. [Internet; Accedido 30-noviembre-2022].
- [10] jsoup. jsoup: Java html parser. <https://jsoup.org/>. [Internet; Accedido 16-noviembre-2022].
- [11] Sergio Koller. ¿qué es el web scraping? <https://seranking.com/es/blog/web-scraping/>, 2021. [Internet; Accedido 12-noviembre-2022].
- [12] Lucid. Lucidchart. <https://www.lucidchart.com/pages/es>. [Internet; Accedido 20-diciembre-2022].
- [13] Moodle. Acerca de moodle. https://docs.moodle.org/all/es/Acerca_de_Moodle. [Internet; Accedido 04-enero-2023].
- [14] Moodle. Moodle core apis. https://docs.moodle.org/dev/Core_APIs. [Internet; Accedido 09-noviembre-2022].
- [15] SEOESTUDIOS. Qué es backend y por qué es tan importante para tu web. <https://www.seoestudios.es/que-es-backend-web/>, 2020. [Internet; Accedido 30-noviembre-2022].
- [16] Piotr Stefaniak. ¿qué es back end y front end? <https://descubrecomunicacion.com/que-es-backend-y-frontend/>. [Internet; Accedido 30-noviembre-2022].
- [17] UC3M. Overleaf - editor online latex. <https://www.uc3m.es/sdic/servicios/overleaf>. [Internet; Accedido 7-octubre-2022].
- [18] Wikipedia. Latex. <https://es.wikipedia.org/wiki/LaTeX>. [Internet; Accedido 7-octubre-2022].