



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**GII 20.09 Herramienta web
repositorios de TFGII**



Presentado por Diana Bringas Ochoa
en Universidad de Burgos — 8 de julio
de 2021

Tutor: Álgvar Arnaiz González y Carlos López
Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Álgvar Arnaiz González, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos. Y, D. Carlos López Nozal, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que la alumna D. Diana Bringas Ochoa, con DNI 79131451-C, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Gestor de TFG 2021.

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, 8 de julio de 2021

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Álgvar Arnaiz González

D. Carlos López Nozal

Resumen

El proyecto propone evolucionar la aplicación web de Gestor de trabajos de fin de grado, una aplicación web 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. Usa como lenguaje de programación Java y emplea componentes gráficos del framework, Vaadin.

Se requiere de mejoras en seguridad, la autenticación de usuarios, en el diseño gráfico con Vaadin y en la obtención de los datos, con la adición de la posibilidad de emplear un nuevo tipo de fichero de datos.

También se solicita la realización de métricas mediante el análisis de la calidad del código en una plataforma online como SonarCloud.

Descriptores

Aplicación web, Java, Vaadin, trabajo de fin de grado (TFG), gestor de proyecto, Moodle, Fillo, Metodología ágil, Spring Boot, SonarCloud, Maven, ApexChart, Heroku ...

Abstract

The project proposes to update the final degree project manager web application, which is in charge of managing the offer, search, assignment and evaluation processes of Computer Engineering career final projects career of the University of Burgos. It uses Java as the main programming language and graphical components of the Vaadin Framework.

It requires improvements in security, like the user authentication, graphic design using Vaadin and data collection, such as the use of a new type of data file.

It are also requested to do metrics by analyzing the quality of the code on an online platform such as SonarCloud.

Keywords

Web application, Java, Vaadin, final degree project, project manager, Moodle, Fillo, Agile methodology, Spring Boot, SonarCloud, Maven, ApexChart, Heroku ...

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>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. Calidad del Software	6
Técnicas y herramientas	9
4.1. Lenguajes de programación	9
4.2. Herramientas de desarrollo	10
4.3. Herramientas de documentación	14
4.4. Gestión del proyecto y control de versiones	15
Aspectos relevantes del desarrollo del proyecto	19
5.1. Gestión del proyecto mediante metodología ágil	19

5.2. Métricas -Análisis de la calidad del código	20
5.3. Integración de la API Fillo	20
5.4. Migración a Vaadin 14	21
5.5. Autenticación de usuarios con UbuVirtual	23
Trabajos relacionados	25
6.1. GII15.9 Gestión de trabajos fin de grado v2.0	25
6.2. GESTFG - Plataforma de Gestión y Evaluación de trabajos de Fin de Grado y Master	26
Conclusiones y Líneas de trabajo futuras	27
7.1. Conclusiones	27
7.2. Líneas de trabajo futuras	28
Bibliografía	29

Índice de figuras

4.1. Esquema del proceso basado en Scrum [20]	16
5.2. Gráfica de ejemplo de la aplicación	22
5.3. Ejemplo tabla con la descripción de proyectos - Proyectos Activos	23
5.4. Ejemplo tabla con la descripción de proyectos ampliada - Pro- yectos Activos	23

Índice de tablas

Introducción

El proyecto se basará en mejorar la actual aplicación del Gestor de TFG, utilizada en el grado de Ingeniería Informática. La aplicación actual se trata de una actualización previa del TFG denominado **GII15.9 Gestión de trabajos fin de grado v2.0**.

La aplicación requería de un login autenticado para la verificación de los usuarios en el acceso a la vista de la actualización de datos. Por lo que en la mejora se integró un **login** para la comprobación de las credenciales y los permisos de acceso mediante **UbuVirtual**.

Otra carencia que padecía era la falta de **compatibilidad con ficheros con múltiples hojas**, lo cual permitiría subir todos los datos de las vistas a la vez. Este problema fue resultado con la integración de una API de Excel, denominada **Fillo**, que permitía la obtención de los datos de un fichero de varias hojas de datos a través de lenguaje SQL.

El proyecto estaba codificado con **versiones obsoletas**, como en el caso Vaadin, una plataforma para el desarrollo de aplicaciones Java. Estos problemas fueron resueltos en esta mejora, reemplazando la gran mayoría de elementos de la aplicación por otros más nuevos y **actualizando las versiones de Vaadin y Java**. Con la actualización de los componentes se logró, a su vez, un mejor diseño de la aplicación.

1.1. Estructura de la memoria

La memoria consta de los siguientes apartados:

- **Introducción:** presentación de la solución propuesta en el proyecto.

- **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 metodológicas y las herramientas de desarrollo empleadas para desarrollar el proyecto.
- **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 o master (TFG/TFM).
- **Conclusiones y líneas de trabajo futuras:** resolución obtenida al concluir el proyecto y descripción de posibles futuras líneas de trabajo o mejoras.

Objetivos del proyecto

En este apartado se detallan los objetivos del proyecto.

2.1. Objetivos generales

- Permitir subir al usuario la información en un nuevo formato con múltiples hojas con datos.
- Mejorar el diseño gráfico de la aplicación.
- Posibilitar la validación de los usuarios a través del correo de la Universidad para el acceso a la vista de la actualización de los datos de la aplicación web.
- Emplear la metodología ágil para realizar el seguimiento del desarrollo del proyecto.

2.2. Objetivos técnicos

- Incorporar una API o plugin para incorporar otro tipo de datos (XLS).
- Integrar un login autenticado a través del correo de la Universidad de Burgos.
- Utilizar GitHub para llevar a cabo el seguimiento del proyecto y control de versiones.
- Crear métricas de la calidad del código de los TFG en una plataforma en la nube.

- Crear la nueva capa de datos asociada al nuevo tipo de datos empleando patrones de diseño.
- Actualizar la versión de Vaadin y de Java empleadas en la aplicación.

Conceptos teóricos

Se explicarán algunos términos importantes para la comprensión del proyecto.

3.1. *Framework*

Un *framework* [18], también denominado marco de trabajo, es una estructura conceptual y tecnológica para facilitar el desarrollo de aplicaciones web mediante un soporte basado en programas, bibliotecas, lenguajes, herramientas, prácticas, criterios...

3.2. *Frontend*

El *frontend* [5] es la parte de una aplicación web o programa que interactúa de forma directa con el usuario. Algunos de los lenguajes más usados para el desarrollo de frontend son: JavaScript, HTML y CSS.

3.3. *Backend*

El *backend* [5] es la parte que se relaciona con la capa de datos, es decir, la base de datos y el servidor. Los usuarios no pueden acceder a ella debido a que contiene lógica de la aplicación. Para el desarrollo del backend se emplean lenguajes como Java y Python.

3.4. Desarrollo *full stack*

El desarrollo *full stack* es la unión del desarrollo del lado del cliente, *frontend*, y el lado del servidor, *backend*.

3.5. Calidad del Software

Hace referencia a una serie de especificaciones que debe cumplir un producto para cumplir con los requisitos establecidos, es decir, lograr la satisfacción del cliente. A pesar de ser un concepto muy amplio y subjetivo se puede calcular empleando estándares como por ejemplo, el **ISO** (*International Organization for Standardization*) el cual describe la calidad como una serie de propiedades y características que debe cumplir un producto o servicio para cumplir con las necesidades del usuario o cliente. Se tendrá en cuenta la etapa del desarrollo del proyecto para determinar la calidad de un proyecto.

Para garantizar la calidad del producto se emplean estándares de calidad. Tanto a **nivel de proceso**, basados en la realización de acciones frecuentemente durante el desarrollo del software, como en el **producto final**, como podrían ser los estándares de documentación y de código.

Para realizar la evaluación de la calidad del software se evalúan, generalmente, empleando los puntos:

- **Funcionalidad:** se define como la capacidad de un producto software para satisfacer las necesidades descritas en circunstancias normales.
- **Fiabilidad:** habilidad del software para mantener el nivel de prestaciones al usarse en condiciones específicas.
- **Seguridad:** se define como la capacidad de un producto software para satisfacer las necesidades descritas en circunstancias normales.
- **Usabilidad:** cualidad del software para dar facilidad de uso y aprendizaje al usuario.
- **Mantenibilidad:** habilidad de un producto software, bajo determinadas condiciones de uso, de conservarse su estado y sufrir modificaciones.
- **Eficiencia:** capacidad para cumplir con las especificaciones con la menor cantidad de recursos posible.
- **Portabilidad:** propiedad para adaptarse a cualquier sistema o entorno.

En el caso de la evaluación del proceso de creación del software, una posible garantía de la calidad es la **gestión ágil**, aunque solo es recomendable en proyectos pequeños. Consiste en la realización de pruebas sobre el código, durante la realización del desarrollo, para la detección de defectos o errores con la mayor brevedad posible. Es posible automatizar este proceso a través de ciertas herramientas como la **Integración continua**.

Técnicas y herramientas

Se van a presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. En su mayoría se ha optado por elegir herramientas usadas previamente, como es el caso de: GitHub, Eclipse y GitHub Desktop.

4.1. Lenguajes de programación

Java

Java [17] es un lenguaje de programación orientado a objetos usado en el desarrollo de aplicaciones. Puede ser empleado tanto por sistemas operativos Unix como Windows debido a que es independiente de la plataforma. Consta de dos partes:

- JRE (*Java Runtime Environment*): es el entorno de ejecución o máquina virtual en la que se ejecuta las aplicaciones en Java. En la versión anterior de la aplicación se empleaba Java 8 pero, con la migración de Vaadin 14 se permitió actualizar a Java 11. En Java 11 la estructura cambió a una plataforma modular y ya no se proporciona el jre.
- JDK (*Java Development Kit*): es el API que distribuye Java para desarrollar aplicaciones empleando el lenguaje de programación. Anteriormente se empleaba Java 8 pero se actualizó a la versión 11, concretamente 11.0.1.0 de jdk.

En la versión anterior se empleaba Java 8 pero, en este proyecto se llevará a cabo una migración a Vaadin 14 para poder actualizar la versión de **Java a la 11**.

CSS

CSS (Hojas de Estilo en Cascada o, en inglés, *Cascading Stylesheets*) [15] es un lenguaje de estilos empleado para el diseño gráfico de componentes o documentos en HTML.

SQL

SQL (Lenguaje de consulta estructurada o, en inglés, *Structures Query Language*) [12] es un lenguaje de programación empleado en el tratamiento de datos y la relación entre ellos. Es usado tanto en la programación como en el diseño y administración de los datos, empleándose en gran medida en la recuperación y almacenamiento de la información en las Bases de datos.

XML

XML (Extensible Markup Language) [6] es un lenguaje marcado que describe el conjunto de reglas para la codificación de documentos. Es simple, general y de facilidad de uso y, por lo tanto, se utiliza para varios servicios web.

4.2. Herramientas de desarrollo

Maven

Maven [8] es una herramienta software con una arquitectura basada en plugins, desarrollada por Apache Software Foundation (ASF), usada para gestionar y construir proyectos en Java.

Se configura el proyecto, a través de un *Project Object Model*(POM) en formato XML (Extensible Markup Language), mediante dependencias con módulos y componentes externos. Además, incluye tareas como la compilación del código, su empaquetado, descarga e instalación de plugins. Existen plugins para trabajar con otros lenguajes como C/C++ y con el Framework .Net.

Vaadin

Vaadin [23] es una plataforma de código abierto para el desarrollo de aplicaciones web con Java. Permite el uso de lenguajes como HTML, CSS y JavaScript, etc. Permite diseñar la interfaz de usuario, medio donde el

usuario interactúa con Java o TypeScript, empleando componentes propios predefinidos de Vaadin.

Requiere emplear un JDK (*Java Development Kit*) y un IDE (Entorno de Desarrollo Integrado) como puede ser [Eclipse](#), [IntelliJ Idea](#) o [NetBeans](#).

Anteriormente se usaba Vaadin 7 pero, como la versión ya no contaba con soporte, se decidió realizar una migración a Vaadin 14. Con la actualización a Vaadin 14, la última versión estable hasta el momento, se pudo emplear JDK 11. Se empleó el IDE Eclipse ya que se había utilizado previamente.

Spring Boot

Spring Boot [9] es un framework de código abierto para Java empleado para el desarrollo de aplicaciones web. Permite ejecutar aplicaciones en formato empaquetado sin necesidad de un servidor web embebido como [Tomcat](#).

Requiere tener instalado un JDK (*Java Development Kit*) de Java, como mínimo la versión 8, y Maven o Gradle. Adicionalmente se puede incluir un servidor como [Apache Tomcat](#).

Eclipse

Eclipse [7] es un IDE (entorno de Desarrollo Integrado) de código abierto y multiplataforma basado en Java. Proporciona diversas herramientas mediante plugins como la conexión con GitHub, PlantUml, etc. Y, múltiples frameworks como Spring boot, Android, Vaadin, Hibernate, JPA, Java EE, entre otros.

Como ya se había utilizado antes no requirió de un periodo de aprendizaje.

Tomcat

[Tomcat](#) [1] es un contenedor de servlets para el despliegue de aplicaciones web.

Fillo

[Fillo](#) [22] es una API (Interfaz de Programación de Aplicaciones) de Excel en Java que permite realizar consultas en lenguaje SQL sobre ficheros en formato xls y xlsx. Soporta consultas, actualizaciones e inserciones en los

ficheros. Tras realizar una exhaustiva búsqueda se optó por emplear Fillo porque era la mejor opción gratuita y de fácil manejo que se encontró.

ApexChart

ApexChart [4] es una librería proveedora de gráficas interactivas para páginas web en formato SVG (Gráfico vectorial escalar). Se trata de un proyecto de código abierto con licencia del MIT y su uso es gratuito en aplicaciones comerciales.

Se emplea en las gráficas de la vista del histórico de la aplicación.

Moodle

Moodle [13] es una plataforma de aprendizaje para proporcionar a profesores, administradores y alumnos un sistema seguro, único y robusto donde crear entornos de aprendizaje personalizados.

Algunas de sus ventajas son: fácil de usar, gratuito, de confianza, multilingües y frecuentemente actualizado.

Firebase

Firebase es un servicio de backend que dispone de SDKs fáciles de usar y bibliotecas de IU ya elaboradas. Incorpora una base de datos online denominada “Firestore Database” y un sistema de autenticación de usuarios denominado “Authentication” de los cuales se hablará en la documentación.

SonarCloud

Es una plataforma gratuita para proyectos de código abierto y compatible con muchos lenguajes de programación, encargada de realizar análisis de calidad del código. También ofrece un servicio de pago para realizar los análisis privados.

Al realizar el análisis de la calidad del código se registran los diferentes tipos de problemas (o *issues*):

- Error o fallo (***bug***): error o fallo del software real.
- **Vulnerabilidad**: puntos débiles de seguridad que pueden emplearse como foco de un ataque.

- Código sucio (***Code Smells***): indica un problema en relación a la mantenibilidad del código puede deberse a qué se estén aplicando malas prácticas en el código o no se estén siguiendo los estándares de los lenguajes. Puede suponer un problema a largo plazo.
- **Código duplicado**: código empleado en diversas partes del código que hace que sea menos eficiente.
- **Cobertura**: se trata del porcentaje del código validado o probado por tests. Cuanto mayor es esta cifra menos probabilidades habrá de que ocasionen errores.

Otros dos conceptos muy importantes en el ámbito de SonarCloud son:

- **Quality Profile**: son colecciones de reglas que se revisan en el análisis en función del lenguaje de programación predeterminado del proyecto.
- **Quality Gate**: condiciones que debe cumplir el proyecto para poder pasar a producción.

Se seleccionó SonarCloud como herramienta gestora de los análisis tras ser recomendada por los tutores.

Heroku

Heroku [21] es una plataforma en la nube con soporte en distintos lenguajes de programación que permite crear y desplegar aplicaciones. Permite añadir nuevas funcionalidades llamados *add ons*, como es el caso de los gestores de bases de datos, por ejemplo PostgreSQL.

El código se ejecuta dentro un contenedor denominado *dyno*, los cuales garantizan la escalabilidad de la aplicación al ejecutarse más o menos dynos en función del número de conexiones que se establezcan. Además, incluye la posibilidad de conectar el código a través de GitHub.

Para emplear Heroku hay que registrarse y crear una cuenta, la cual puede ser:

- Gratuita: incluye las prestaciones básicas como el despliegue de una aplicación. Es la versión que se está empleando.
- De pago: incorpora todas los servicios de Heroku.

Tras una búsqueda y análisis de las plataformas con hosting de despliegue gratuito se escogió Heroku porque proporcionaba más servicios y ventajas que el resto.

4.3. Herramientas de documentación

Se explicaran brevemente los programas empleados para realizar la documentación, tanto para el diseño de diagramas como para la redacción de la propia documentación.

LaTeX

LaTeX [14] es un software libre para la composición de textos con una gran calidad tipográfica. Es empleado en gran medida para la creación de artículos, libros técnicos y tesis.

En el proyecto se utilizó, recomendado por los tutores, **Miktex** como distribuidor de TeX junto con el editor de código abierto y multiplataforma, **TeXstudio**.

Miktex

MikTeX [24] es una distribución de TeX/LaTeX multiplataforma que cuenta con un gestor de paquetes con la capacidad de instalar los componentes faltantes de Internet si fuese necesario.

TeXstudio

TeXstudio [16] es un editor (IDE) de código abierto multiplataforma, similar al editor Texmaker, con la capacidad de editar, dar soporte, resaltar sintaxis, realizar revisiones ortográficas del código en LaTeX y visualizar el documento en un visor en formato pdf.

Se ha empleado para elaborar la documentación del proyecto haciendo uso de la plantilla proporcionada proporcionada.

StartUML

Se trata de una aplicación de escritorio para el diseño de diagramas UML, como es el caso de los diagramas de casos de uso, de clase o de secuencia. Se trabajó con este programa ya que anteriormente se había usado.

Lucid app

Es una plataforma online de diseño de diagramas gratuito y fácil de usar.

PlantUML

PlantUML [19] es una herramienta de código abierto para la creación de diagramas UML, a partir de lenguaje de texto sin formato. Se empleó para los diagramas de clase empleados en la documentación.

4.4. Gestión del proyecto y control de versiones

Para el control de versiones se ha optado por utilizar programas y plataformas ya conocidas.

Metodología Scrum

Scrum [2] se trata de una metodología iterativa e incremental, empleada en el desarrollo y gestión ágil de proyectos. Describe una serie de buenas prácticas para promover la colaboración en equipos, consiguiendo mejores resultados que cumplen con las necesidades del cliente.

Comienza con la creación de una lista denominada *product backlog* que servirá de guía para el desarrollo del producto. Se determinan las tareas (en ingles, *issue*) que deben realizarse en cada intervalo de tiempo, al que se le nombra como *sprint*. Generalmente un *sprint* dura entre una a cuatro semanas. Se podrá añadir características a las tareas como la prioridad, la descripción del proceso, el tiempo estimado que llevará realizarla, el tipo, el responsable o persona asignada para dicha labor, entre otros.

Para supervisar el desarrollo se realizarán reuniones cortas, alrededor de quince minutos, y periódicas, donde se presentará el *feedback* del producto al cliente. El equipo analizará el estado del proyecto y los problemas a futuro o ocasionados previamente. Según el estado en el que se encuentre el proyecto, se decidirá nuevamente una lista de tareas (*product backlog*) para el siguiente *sprint*. Se puede apreciar de forma resumida en la ilustración 4.1.

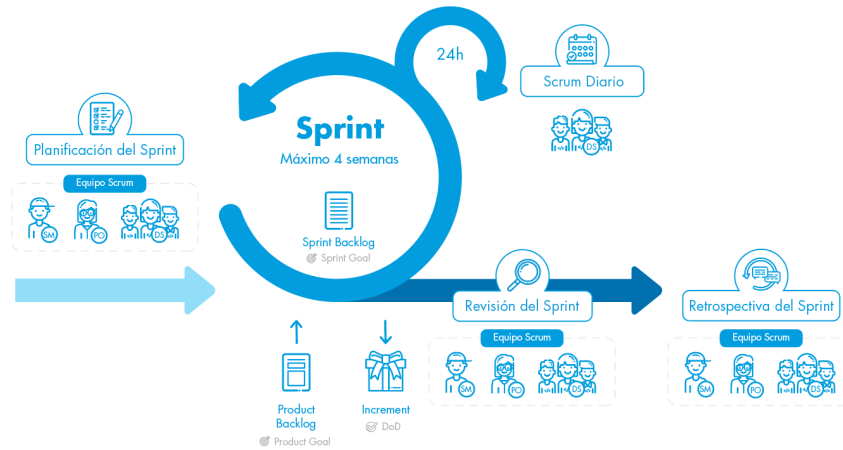


Figura 4.1: Esquema del proceso basado en Scrum [20]

Este proceso continuará hasta la finalización del proyecto obteniendo el producto definitivo.

En el **proyecto del Gestor de TFG 2021**, se integró esta metodología a través de **Github**, donde aproximadamente cada quince días se realizaba una la reunión de seguimiento del TFG con los tutores. En las reuniones se revisaba los cambios introducidos en el anterior *sprint*, los problemas que se pudiesen haber producido y se planteaba las siguientes *issues* que debían desarrollarse en el siguiente *sprint*.

Integración Continua

La CI (Integración continua) [3] es una práctica de desarrollo de software que consiste en la búsqueda de posibles fallos en cada actualización del código del repositorio que se realice. Esta comprobación se realiza a través de la compilación del código fuente y la ejecución de los test. Para conseguir un proceso eficaz se deberán realizar integraciones del código periódicamente para poder detectar errores a la mayor brevedad posible.

Este proceso se realiza durante la fase de desarrollo o integración del software y conlleva un componente de automatización que en este caso se realizará mediante la opción **GitHub Actions** de Github.

GitHub

GitHub [10] es una plataforma de repositorios online colaborativos que permite llevar a cabo la gestión de proyectos y el control de versiones. A través de esta herramienta se realizó el seguimiento del proyecto integrando la **metodología ágil Scrum**.

La principal razón por la que se optó por Github fue porque ya se tenían ciertos conocimiento acerca de su funcionamiento.

ZenHub

ZenHub [25] es una plataforma de gestión de proyectos totalmente integrada en GitHub. Organiza las issues en el tablero *canvas* según su estado: recién creadas, pendientes, en proceso, ya terminadas, etc.

También incluye la posibilidad de generar gráficas para la obtención de información acerca del flujo del trabajo a lo largo del tiempo. Por ejemplo, el diagrama Burndown, empleado en los anexos de la documentación, muestra la comparativa del tiempo estimado para la realización de las tareas y el tiempo real que ha tomado en realizarse.

Se empleará esta herramienta junto con Github para realizar el seguimiento del proyecto empleando la Metodología Ágil Scrum.

GitHub Desktop

Github Desktop [11] simplifica la conexión con el repositorio GitHub sin necesidad de usar la línea de comandos de Git. A través de este programa se pueden realizar *commit* y subirlos al GitHub mediante el comando *push*, bajar los cambios realizados en el repositorio con *pull*, entre otros.

Aspectos relevantes del desarrollo del proyecto

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

5.1. Gestión del proyecto mediante metodología ágil

Se introdujo el uso de la **integración continua (CI)** de forma automática con **GitHub** mediante la opción **GitHub Actions**.

Se creó un flujo de trabajo (*workflows*) que compilaba y ejecutaba los test del código con el fin de verificar que no existían fallos en el código. Se realizaba esta comprobación cada vez que se realizaban actualizaciones en el código, subido al repositorio de GitHub, consiguiendo detectar los errores con mayor brevedad.

También se realizaron **actualizaciones del despliegue de la aplicación** en **Heroku** con cada modificación del código para mantener el estado de la aplicación web desplegada al día. En lugar de usar la aplicación desplegada en local en las reuniones de seguimiento con los tutores, se empleaba la aplicación desplegada porque funcionaba con mayor fluidez y rapidez.

5.2. Métricas -Análisis de la calidad del código

Un requisito era la realización del **análisis de la calidad del código a proyectos realizados en años anteriores** en una plataforma online, **SonarCloud**.

Primero se creó una cuenta en **SonarCloud** y se comenzó a indagar como realizar las mediciones de la calidad en la **documentación de SonarCloud**.

Con los proyectos con lenguajes de programación compatibles con SonarCloud, por ejemplo css, se podría realizar el análisis automático con lo que era mucho más sencillo porque no hacía falta especificar nada acerca del proyecto. Sin embargo, con los lenguajes no soportados como Java se requería especificar dónde se encontraban los archivos fuente y binarios. En la documentación de **SonarCloud acerca del análisis automático** se nombran los lenguajes soportados por la plataforma.

Se analizaron alrededor de 36 proyectos, sin contar los proyectos que no se pudieron realizar por fallos.

5.3. Integración de la API Fillo

Uno de los objetivos era introducir la recuperación de datos con ficheros de múltiples hojas, concretamente en formato ods. Para ello se investigó una alternativa similar al driver empleado para leer los ficheros csv. Primero se probó la opción de **Microsoft Excel JDBC Driver** con el cual se puede leer, escribir y actualizar Excel mediante JDBC. Sin embargo, es una opción es de pago, por lo que fue descartada.

Después se testeó dos drivers gratuitos, **ODFDOM** y **JopenDocument**, los cuales estaban en desuso.

Como no se encontró ninguna solución válida, se decidió con los tutores cambiar el formato del fichero a xls, ya que en este formato existían más opciones. Se volvió a realizar una búsqueda de drivers o APIs para realizar la conexión con ficheros xls y se encontraron varias opciones:

- **Apache POI**: permite, mediante bibliotecas en Java puro, leer y escribir en archivos en formatos de Microsoft Office como Excel. Para verificar su funcionamiento, se incluyó en el proyecto de prueba “Hola-MundoVaadin” y se realizaron diversas pruebas. Aunque funcionaba

bien fue descartado ya que su inclusión requería de rehacer todas las funciones de obtención de datos y no se podría reutilizar código, empleado en la lectura de los ficheros csv.

- **SqlSheet**: driver en JDBC para la conexión con ficheros xls y xlsx basado en Apache POI. Aunque era una buena opción, fue eliminada porque solo permitía hacer consultas sencillas de todo el documento, en lugar de seleccionar los datos por columnas o según una condición.
- **Fillo**: es una API(Interfaz de Programación de Aplicaciones) de Excel en Java que permite realizar consultas en lenguaje SQL sobre ficheros en formato xls y xlsx. Para testearlo se usó el proyecto de prueba “HolaMundoVaadin” y tras verificar su funcionamiento se incluyó en el proyecto principal. Se escogió esta opción porque exigía menos modificaciones y la posibilidad de reutilizar código.

Para realizar la integración de **Fillo** se separó el código referente a la lectura de los datos en dos partes, una para cada tipo de fichero que la aplicación permite. Esto ocasionó la creación de nuevos test para la comprobación de la obtención de la información en el nuevo tipo de fichero.

5.4. Migración a Vaadin 14

La aplicación empleaba Vaadin 7 pero, cómo ya no contaba con soporte y requería usar Java 8 se decidió actualizar Vaadin a la versión 14, la última versión estable.

Para realizar la migración se intentó incorporar **MPR**(siglas en inglés, *Multiplatform Runtime*) que ejecuta la aplicación original, en Vaadin 7, dentro de una aplicación en **Vaadin 14**. Para ello, se siguió la [documentación de MPR en Vaadin](#) y se tomó como ejemplo el [repositorio de demostración](#) mencionado en la documentación.

Tras intentar realizar la migración mediante **MPR**(siglas en inglés, *Multiplatform Runtime*) y no conseguir resultados, se desistió y se comenzó a realizar la **migración de cero a Vaadin 14**. Se descargó uno de los proyectos de ejemplo de Vaadin 14, para tomarlo como referencia.

La actualización de los componentes y la navegación conllevó realizar una búsqueda y aprendizaje de los componentes en Vaadin 14, ya que se apreciaba un gran cambio.

Otro apartado en el que se apreció un gran cambio fue en la navegación de la aplicación que se incluyó **Spring Boot** porque ofrecía una manera sencilla y rápida de ejecutar la aplicación sin necesidad de añadir un servidor web embebido.

La migración requirió más cambios de los esperados y como consecuencia se invirtió mucho tiempo en la tarea.

Mejoras de componentes de la aplicación

Con el cambio de versión se realizó una **mejora de los componentes**, como por ejemplo el reemplazo del elemento para realizar los gráficos por **ApexCharts** consiguiendo gráficos interactivos y atractivos 5.2.

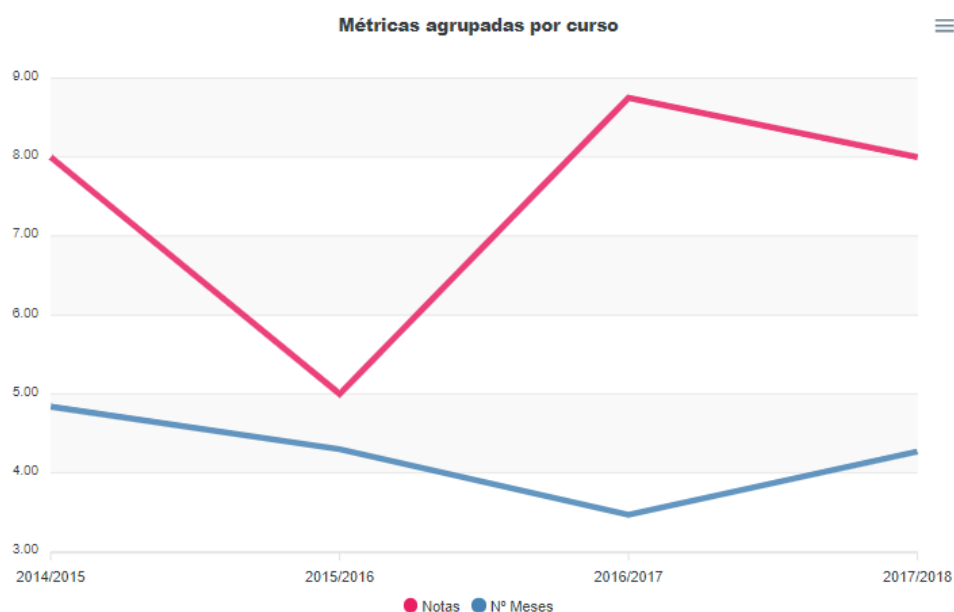


Figura 5.2: Gráfica de ejemplo de la aplicación

La mejora de las **tablas** e introducción de nuevas funcionalidades como los rankings de las calificaciones 5.3 y la posibilidad de ampliar las filas para ver con más detalle su contenido 5.4.

Descripción de proyectos

Título	Tutor/es	Nº Alumnos	Fecha Asignación	Fecha Presentación	Ranking Percentil...	Ranking Total	Ranking por curso
Título corto 1	Tutor1	2	2016-11-01	2017-07-01	B	2	2
Título corto 2	Tutor2, Tutor6, Tutor3	1	2016-10-01	2017-07-01	A	1	1
Título corto 3	Tutor3	3	2015-12-02	2016-07-02	E	6	2
Título corto 4	Tutor4, Tutor1	1	2015-10-03	2016-07-03	D	5	1
Título corto 5	Tutor5	2	2015-11-04	2016-07-04	E	7	3
Título corto 6	Tutor2	1	2017-09-05	2018-07-05	A	1	1
Título corto 7	Tutor4	1	2017-10-06	2018-07-06	C	4	2
Título corto 8	Tutor5	1	2014-11-07	2015-07-07	C	3	1

Figura 5.3: Ejemplo tabla con la descripción de proyectos - Proyectos Activos

Título	Tutor/es	Nº Alumnos	Fecha Asignación	Fecha Presentación	Ranking Percentil...	Ranking Total	Ranking por curso
Título corto 1	Tutor1	2	2016-11-01	2017-07-01	B	2	2

Título: Título corto 1
 Tutor/es: Tutor1
 Nº Alumnos: 2
 Fecha Asignación: 2016-11-01
 Fecha Presentación: 2017-07-01
 Ranking Percentiles: B
 Ranking Total: 2
 Ranking por curso: 2

Figura 5.4: Ejemplo tabla con la descripción de proyectos ampliada - Proyectos Activos

5.5. Autenticación de usuarios con UbuVirtual

Uno de los requisitos era realizar un login que permitiera la autenticación de los usuarios con el correo de la Universidad de Burgos. Para ello, se recopiló posibles herramientas que podían usarse para realizar la conexión y verificación del usuario.

Primero se intentó realizar **la conexión a través de Microsoft** como se indica en su **documentación sobre Azure** pero no se consiguió.

Después se probó con **Firestore**, con la opción de “**Authentication**” desde la cual se puede añadir usuarios y gestionar sus permisos. El problema era que requería de una persona que introdujera los datos y de realizar un proceso de codificación y decodificación de las contraseñas, por lo que no era viable. Con el caso de la base de datos online que tiene Firestore, “**Firestore**” ocurría lo mismo por lo que también se descartó.

Se quería introducir un sistema que no necesitase de una persona para gestionarlo, por tanto, se optó por la autenticación mediante la instancia del **Moodle de UbuVirtual**.

Como consecuencia de esta modificación se tuvo que añadir mucho código y aprender cómo realizar la conexión y obtención de la información de Moodle, ya que nunca se había trabajado con esta plataforma.

Trabajos relacionados

Se nombrarán algunos proyectos y aplicaciones similares o relacionados con la gestión de trabajos de fin de grado o master (TFG/TFM).

6.1. GII15.9 Gestión de trabajos fin de grado v2.0

Es el proyecto que se propone para mejorar consiste en una aplicación web para la gestión de TFG de la Universidad de Burgos. Emplea como lenguaje de programación Java y Vaadin 7 como plataforma de código abierto para la interfaz web.

Se trata de una actualización del TFG denominado **Revisiones automáticas de calidad de proyectos con SonarQube**. Se introdujo la lectura de ficheros csv con el driver “CsvDriver”, el análisis local de los trabajos de fin de grado en **SonarQube** y la generación de las vistas de la aplicación empleando los datos obtenidos de los csv.

El código fuente se puede encontrar en el **repositorio de GitHub**. Para ejecutar el código se abrirá un terminal dentro del proyecto principal llamado “**sistinf**” y se instalarán las dependencias con “*mvn install*”.

Después se compilará y empaquetará el código en un war con el comando “*mvn clean package*” y, por último, se desplegará con el servidor HTTP, **Jetty** con “*mvn jetty:run-war*”.

Si en lugar de desplegar el proyecto se desea ejecutar a través de un IDE, se realizará únicamente la instalación de dependencias y la ejecución de la app con “*mvn jetty:run*”.

La aplicación se desplegará en la url por defecto <http://localhost:8080/>.

6.2. GESTFG - Plataforma de Gestión y Evaluación de trabajos de Fin de Grado y Master

Es un sistema de gestión de TFG y TFM empleado en la Escuela Técnica Superior de Ingenierías Informática y Telecomunicaciones de Granada. Se empleó el framework [Django](#). La plataforma administra la información relacionada con los TFGs, usuarios, las diferentes fases (asignación, evaluación), interacción entre los miembros asignados a un TFG (alumnos, tutores y tribunal), notificaciones vía correo electrónico, evaluación de los TFG, entre otras funcionalidades.

El proyecto se encontraba almacenado en [Docker](#), plataforma online de almacenaje y despliegue de proyectos en contenedores.

Con el fin de probar el código del trabajo, ubicado en el repositorio [gestion-tfg](#), se ha descargado la imagen del docker y se han realizado los pasos descritos en la [propia documentación del proyecto](#). Pero, al ejecutar dicha imagen da error. Se ha intentado realizar con otros métodos, recomendados por los tutores, pero no fue posible.

Anteriormente se encontraba desplegado a través de GitHub pero ya no se encuentra disponible, debido seguramente a la falta de mantenimiento, ya que la última fecha de modificación es de hace cinco años.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Una vez finalizado el trabajo se puede ver una evolución en cuanto a las herramientas empleadas y el manejo de las metodologías ágiles. Al realizar un seguimiento del proyecto a través de herramientas como GitHub y ZenHub que ha permitido un mejor control de las tareas.

Por otro lado, al ser un proyecto que propone una mejora de una **aplicación ya existente** implica la **comprensión y aprendizaje del funcionamiento** del mismo. Al comienzo del desarrollo del proyecto se empleó mucho tiempo en conocer el código y aprender las nuevas herramientas que se iban a emplear, entre las cuales destaca Vaadin. Gracias a la cual, se ha logrado desarrollar los componentes de la aplicación sin grandes conocimientos en lenguajes de frontend como CSS o JavaScript. Otro punto a tener en cuenta es el empleo de Maven y Spring Boot que facilitan la configuración de las dependencias.

Al tener que **migrar de versión de Vaadin**, se tuvo que volver a aprender el funcionamiento de la navegación de Spring Boot y de los componentes de Vaadin 14, lo que requirió de mucho tiempo de desarrollo.

Otro punto donde se necesitó bastante tiempo fue en la **búsqueda de una API para la obtención de los datos de los ficheros con múltiples hojas**. A causa de la falta de opciones válidas que fuesen gratuitas. Se decidió cambiar de formato a xls para poder encontrar propuestas que pudiese servir.

El análisis de la **calidad del código** de los proyectos con lenguajes de programación soportados fue sencilla de realizar, pero los proyectos con lenguajes no compatibles requirieron de la comprensión de la estructura de los ficheros a través de la documentación del TFG.

7.2. Líneas de trabajo futuras

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

- Añadir la **posibilidad de solicitar la asignación de un TFG** a través de la aplicación web. Al escoger un TFG le llegaría un mensaje a los tutores designados, los cuales se podrían en contacto con el alumno interesado.
- Modificar la columna de la tabla del Histórico, correspondiente al **ranking** de percentiles, para que **varíe el color del texto** según el valor de la celda. Introduciendo colores similares a los que se usa en SonarCloud, verde para el mejor resultado y rojo para el peor.
- 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.
- Incluir la **internacionalización** de la aplicación.

Bibliografía

- [1]
- [2] Blog Proyectos Agiles. Qué es scrum. <https://proyectosagiles.org/que-es-scrum/>. [Internet; Accedido 01-julio-2021].
- [3] Amazon. ¿qué es la integración continua? <https://aws.amazon.com/es/devops/continuous-integration/>. [Internet; Accedido 01-julio-2021].
- [4] ApexChart. Apexchart. <https://apexcharts.com/1>. [Internet; Accedido 03-julio-2021].
- [5] Nicole Chapaval. Qué es frontend y backend. <https://platzi.com/blog/que-es-frontend-y-backend/>, 2018. [Internet; Accedido 24-noviembre-2020].
- [6] Ivan de Souza. Xml: ¿qué es y para qué sirve este lenguaje de marcado? <https://rockcontent.com/es/blog/que-es-xml/>. [Internet; Accedido 25-junio-2021].
- [7] Inc. Eclipse Foundation. Eclipse. <https://www.eclipse.org/>. [Internet; Accedido 03-julio-2021].
- [8] The Apache Software Foundation. Maven. <https://maven.apache.org/>. [Internet; Accedido 24-noviembre-2020].
- [9] Domingo Gallardo. Introducción a spring boot. <https://domingogallardo.github.io/practicas-mads/01-intro-spring-boot/intro-spring-boot.html>. [Internet; Accedido 02-julio-2021].

- [10] Inc. GitHub. Github. <https://github.com/>. [Internet; Accedido 24-noviembre-2020].
- [11] Inc. GitHub. Github desktop. <https://desktop.github.com/>. [Internet; Accedido 22-noviembre-2020].
- [12] Microsoft. Access sql: conceptos básicos, vocabulario y sintaxis. <https://support.microsoft.com/es-es/office/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671>. [Internet; Accedido 05-julio-2021].
- [13] Moodle. Acerca de moodle. https://docs.moodle.org/all/es/Acerca_de_Moodle#:~:text=Moodle%20es%20una%20plataforma%20de,crear%20ambientes%20de%20aprendizaje%20personalizados.&text=Nota%20del%20traductor%3A%20Si%20necesita,Moodle%20gratuito%2C%20vea%20Moodle%20gratuito. [Internet; Accedido 03-julio-2021].
- [14] Sergio Luján Mora. ¿qué es latex? <http://desarrolloweb.dlsi.ua.es/cursos/2015/herramientas-investigacion/que-es-latex>. [Internet; Accedido 03-julio-2021].
- [15] Mozilla and individual contributors. Css. <https://developer.mozilla.org/es/docs/Learn/CSS>. [Internet; Accedido 25-junio-2021].
- [16] Noel. ¿qué es texstudio y cómo se instala? <https://linux.com/que-es-texstudio-y-como-se-instala/>. [Internet; Accedido 04-julio-2021].
- [17] Oracle. Java. https://java.com/es/download/help/whatis_java.html. [Internet; Accedido 24-noviembre-2020].
- [18] 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.,2018](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 24-noviembre-2020].
- [19] Adrián Peláez Paniagua. Plantuml – dibuja diagramas uml de forma sencilla. <https://www.adictosaltrabajo.com/2015/12/18/tutorial-plantuml-dibuja-diagramas-de-forma-sencilla/>. [Internet; Accedido 01-julio-2021].

- [20] Miquel Rodríguez. Scrum: el pasado y el futuro. <https://netmind.net/es/scrum-el-pasado-y-el-futuro/>. [Internet; Accedido 01-julio-2021].
- [21] Salesforce.com. Heroku. <https://www.heroku.com/home>. [Internet; Accedido 14-febrero-2021].
- [22] Softwaretestinghelp. How to read or write data from excel sheet in selenium web driver. <https://www.softwaretestinghelp.com/read-write-data-from-excel-sheet-in-selenium-web-driver/>. [Internet; Accedido 04-julio-2021].
- [23] Vaadin. Vaadin. <https://vaadin.com/>. [Internet; Accedido 24-noviembre-2020].
- [24] Wikipedia. Miktex. <https://en.wikipedia.org/wiki/MiKTeX/>. [Internet; Accedido 03-julio-2021].
- [25] ZenHub. Zenhub. <https://www.zenhub.com/>. [Internet; Accedido 22-noviembre-2020].