



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Asistente de buenas prácticas
para proyectos en GitHub**



Presentado por Lucas Olmedo Díez
en Universidad de Burgos — Junio de 2025
Tutor: Carlos López Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Carlos López Nozal, profesor del departamento de nombre departamento,
área de nombre área.

Expone:

Que el alumno D. Lucas Olmedo Díez, con DNI 71306075A, ha realizado el
Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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, Junio 2025

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Carlos López Nozal

D. nombre co-tutor

Resumen

Las prácticas ágiles han demostrado ser fundamentales en el desarrollo de software, ya que promueven la entrega iterativa, la mejora continua y una mejor colaboración entre los desarrolladores. Su aplicación es observable en los repositorios open source de plataformas como GitHub, que facilitan la gestión del código, la documentación y la colaboración entre desarrolladores. En el contexto académico, GitHub es ampliamente utilizado por los estudiantes para la gestión de sus Trabajos de Fin de Grado (TFG). Sin embargo, muchos enfrentan dificultades para aplicar correctamente las prácticas ágiles en sus repositorios, lo que puede afectar la calidad y eficiencia de su desarrollo. El objetivo de este trabajo es ayudar a los estudiantes a aplicar prácticas ágiles en sus repositorios de GitHub mediante una aplicación web que evalúa su grado de adopción. Para ello, la aplicación analiza el uso del repositorio según consultas y reglas basadas en prácticas ágiles, comparando los resultados con valores de referencia fijos o con repositorios seleccionados como modelo. De esta manera, la herramienta facilita la identificación de áreas de mejora y fomenta la adopción de buenas prácticas en la gestión y desarrollo de proyectos académicos en GitHub. La aplicación web está escrita en Python y Angular y se puede acceder a un despliegue (añadir url) y a su código (añadir url).

Descriptores

Métricas de calidad, proceso de desarrollo de software, metodologías ágiles, integración continua, documentación continua, GitHub, análisis de repositorios, aplicaciones web.

Abstract

Agile practices have proven to be essential for software development. This is because they promote iterative delivery, continuous improvement, and better collaboration between developers. Its usage is visible in open source repositories from platforms like GitHub, which make code administration, documentation, and collaboration between developers much easier. In the academic context, GitHub is heavily used by students for their TFG's administration. However, many of them face difficulties to apply correctly agile methodology in their projects, which can affect their developments' quality and efficiency. The objective of this project is to help students apply agile practices in their GitHub repositories by a web app that evaluates the usage of agile methodology. To achieve that, the app analyzes the use of the repository using queries and rules based on agile practices, comparing results with fixed reference values or repositories selected as a comparison source. This way, the tool helps the user identify areas of improvement, and encourages the user adopting agile practices in their academic GitHub's projects' administration and development. The web app is written in Python and Angular, and can be accessed to its deployment (URL) and its code (URL).

Keywords

Evolution metrics, software developing process, agile methodology, continuous integration and documentation, GitHub, repository analysis, web applications.

Índice general

Índice general	iii
Índice de figuras	iv
Índice de tablas	v
1. Introducción	1
2. Objetivos del proyecto	3
3. Conceptos teóricos	5
3.1. Secciones	5
3.2. Referencias	5
3.3. Imágenes	6
3.4. Listas de ítems	6
3.5. Tablas	7
4. Técnicas y herramientas	9
5. Aspectos relevantes del desarrollo del proyecto	11
6. Trabajos relacionados	13
7. Conclusiones y Líneas de trabajo futuras	15
Bibliografía	17

Índice de figuras

3.1. Autómata para una expresión vacía	6
--	---

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	7
---	---

1. Introducción

El desarrollo de software es un proceso complejo que involucra múltiples factores técnicos y organizativos. En el ámbito técnico, es fundamental garantizar el cumplimiento de los requisitos funcionales y no funcionales, tales como mantenibilidad, escalabilidad, eficiencia y calidad del código[3]. A nivel organizativo, la gestión efectiva de equipos de desarrollo, la planificación de tareas y la optimización del tiempo y recursos son aspectos clave para el éxito de un proyecto. Para abordar esta complejidad, se han desarrollado metodologías y herramientas que facilitan la gestión del proceso de desarrollo, promoviendo la colaboración y la mejora continua.

Las metodologías ágiles, como Scrum[6], Kanban[5] y eXtreme Programming[1], han demostrado ser eficaces para gestionar proyectos de software de manera flexible e iterativa. Estas metodologías enfatizan la entrega incremental de valor, la integración continua, la colaboración entre equipos y la adaptación constante a los cambios en los requisitos del proyecto. En este contexto, la gestión de repositorios de código, como GitHub y GitLab, desempeña un papel crucial al proporcionar funcionalidades avanzadas para el control de versiones, la gestión de issues, la revisión de código y la automatización de flujos de integración y despliegue continuo.

Sin embargo, la gran cantidad de datos generados en los repositorios de software dificulta la evaluación de la calidad del desarrollo y la documentación de este, así como la identificación de áreas de mejora. En respuesta a esta necesidad, este TFG propone el desarrollo de una aplicación web que analiza la información de un repositorio de GitHub y, mediante la aplicación de buenas prácticas ágiles y métricas de calidad, ofrece asistencia a los desarrolladores para mejorar su proceso de trabajo.

El sistema evaluará aspectos clave del desarrollo, como la velocidad de trabajo (frecuencia de commits e issues cerradas), la documentación continua, la calidad de la documentación de los commits e issues (uso de descripciones, etiquetas, y demás herramientas de documentación) y la implementación de pruebas automatizadas. Para ello, se utilizarán métricas cuantitativas que permitan visualizar el rendimiento del equipo y la evolución del proyecto a lo largo del tiempo. Además, el sistema proporcionará recomendaciones basadas en prácticas ágiles establecidas en fuentes como "Subway Map to Agile Practices" de Agile Alliance, tales como integración continua, revisión de código, retrospectivas y gestión visual del trabajo.

Este proyecto busca facilitar la adopción de metodologías ágiles y mejorar la calidad del software y su documentación mediante la automatización del análisis de repositorios. A través de este enfoque, los equipos de desarrollo podrán optimizar sus procesos, mejorar la colaboración y garantizar la entrega de software más robusto y bien documentado.

2. Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

3. Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX ¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [7]. Para citar webs, artículos o libros [4], si se desean citar más de uno en el mismo lugar [2, 4].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

■

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

4. Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

6. Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

7. Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2nd edition, 2004.
- [2] Zachary J Bortolot and Randolph H Wynne. Estimating forest biomass using small footprint lidar data: An individual tree-based approach that incorporates training data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(6):342–360, 2005.
- [3] International Organization for Standardization (ISO). Iso/iec 25000, software engineering - software product quality requirements and evaluation (square), 2005.
- [4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [5] Paul Roe and Tony Richards. Applying kanban in software engineering: A case study. *Journal of Software Engineering Research and Development*, 5(1):12–25, 2017.
- [6] Jeff Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. Currency, 2014.
- [7] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].