



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Evaluación de simulaciones de
proyectos en GitHub
orientadas a docencia.**



Presentado por Licinio Fernández Maurelo
en Universidad de Burgos — 26 de octubre
de 2022

Tutor: Carlos López Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Licinio Fernández Maurelo, con DNI dni, 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, 26 de octubre de 2022

Vº. Bº. del Tutor:

D. Carlos López Nozal

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	5
Conceptos teóricos	7
3.1. Conceptos de la gestión de proyectos ágiles utilizados en el laboratorio virtual.	7
3.2. Conceptos de control de versiones utilizados en el laboratorio virtual	9
3.3. Secciones	10
3.4. Referencias	10
3.5. Imágenes	11
3.6. Listas de items	11
3.7. Tablas	11
Técnicas y herramientas	13
4.1. Arquitectura hexagonal	13
4.2. Integración continua	13
Aspectos relevantes del desarrollo del proyecto	15
Trabajos relacionados	17

Conclusiones y Líneas de trabajo futuras	19
Bibliografía	21

Índice de figuras

1.1. Gestión de tareas de un proyecto software: extracto de información para la simulación	2
1.2. Gestión de tareas de un proyecto software: extracto de rúbrica para la evaluación	3

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto corta	11
3.2. Herramientas y tecnologías utilizadas en cada parte del proyecto	12

Introducción

En la actualidad, la enseñanza a distancia goza de gran importancia debido a ventajas inherentes como la deslocalización de profesores y alumnos, la libertad de agenda de los alumnos a la hora de seguir la formación y el uso eficiente de los recursos educativos que permite a los centros de formación ampliar su capacidad de alumnado al no estar sujetos a limitaciones propias de la enseñanza presencial como aulas y laboratorios.

Sin embargo, la enseñanza a distancia de materias relacionadas con la ciencia, la tecnología y la ingeniería presenta dificultades derivadas de la propia naturaleza de estas disciplinas: a menudo requieren prácticas en el laboratorio para dotar al alumno de las competencias necesarias . [2] menciona cómo, históricamente, algunas de las necesidades especiales de los estudios universitarios de ingeniería no han sido bien cubiertas por los métodos de enseñanza online.

La emergencia de las tecnologías de la información ha permitido el desarrollo de laboratorios virtuales. Su acceso online y el software que los soporta permite a los alumnos una reproducción cada vez más fiel del trabajo realizado en el laboratorio.

Como [6] señala, los laboratorios virtuales son esenciales para que los alumnos adquieran conocimientos prácticos que se convertirán en activos fundamentales en su carrera profesional. En el desarrollo de laboratorios virtuales, uno de los focos de acción es facilitar la autoevaluación para mejorar la experiencia de aprendizaje de los alumnos y reducir la carga de trabajo de los docentes. Una autoevaluación clara y concisa lleva al alumno a una reflexión sobre sus propios errores, convirtiendo una evaluación formativa en una evaluación formadora [5].

La enseñanza de un modelo ágil de gestión de proyectos mediante GitHub, ZenHub y el sistema de control de versiones Git proporciona el marco de este trabajo de fin de grado. Corresponde a la asignatura Gestión de proyectos - bloque gestión ágil, del tercer curso del grado en ingeniería informática. En esta asignatura se plantean dos actividades a modo de laboratorio virtual:

- Gestión de tareas de un proyecto software. El alumno realiza una simulación de planificación de tareas de un proyecto software.
- Simulación de control de versiones de un proyecto open source. El alumno simula la gestión de versiones de código utilizando un proyecto disponible en un repositorio público.

Los supuestos de ambas actividades proporcionan el detalle paso por paso de la simulación a realizar, como el que se muestra en la figura 1.1 y, además, el alumno cuenta con el repositorio original a modo de referencia.

Sprint 16	17 closed = 7	
	documentation + 4	1 issue con comment
6 días	feature + 2 testing + 4	
	bug	#164 Las imágenes no se posicionan correctamente (enlace + 8 commits)
53 puntos		
de	Todas asignados a	0 issue con tasklist
historia	David	
		3 issues con comments
	9 closed = 2	

Figura 1.1: Gestión de tareas de un proyecto software: extracto de información para la simulación

El producto entregable de cada una de las prácticas consta de capturas de pantalla que constatan la fidelidad de la simulación con respecto al original. Para la evaluación de cada una de las prácticas, existe una rúbrica de evaluación que actualmente es aplicada de forma manual en base al producto entregado.

Valoración del trabajo en equipo en Github	Participación menos del 50% de los miembros del equipo en las tareas 0 points		Participación del 100% de los miembros del equipo en las tareas 2 points
Estimaciones temporales	Existen más del 75% de elementos de Scrum sin asignar tiempos 0 points	Se aplican tiempos a todos los elementos Scrum de la aplicación pero se desvian mucho de la realidad 1 points	Se aplican tiempos a todos los elementos Scrum planificados. Los tiempos se aproximan a la realidad. 2 points

Figura 1.2: Gestión de tareas de un proyecto software: extracto de rúbrica para la evaluación

La evaluación de ambas actividades puede automatizarse en gran medida. La información de la planificación de un proyecto en GitHub y en ZenHub, así como la información del control de versiones de GitHub, es accesible vía API. A partir del contraste de la información del proyecto referencia y el proyecto de la simulación podemos dar contestación a gran parte de los elementos de la rúbrica. Del mismo modo, podemos explotar esta comparación para proporcionar retroalimentación al alumno de aquellos elementos de la simulación que concuerdan o discrepan con el original.

La evaluación automática de las competencias de gestión de tareas y del control de versiones es de utilidad fuera del ámbito académico, por ejemplo, en el proceso de evaluación de competencias de profesionales con esas habilidades en el mundo empresarial, en concreto:

- Procesos de selección.
- Evaluaciones internas de personal de la compañía.

En este trabajo, se abordará la evaluación en el plano académico. Podría haberse abordado desde la perspectiva de un proceso de control de calidad.

Objetivos del proyecto

La realización de este proyecto tiene como objetivo principal el desarrollo de un software que permita automatizar la evaluación y retroalimentación de las siguientes actividades planteadas como laboratorio virtual en la asignatura gestión de proyectos del grado en ingeniería informática:

1. Gestión de tareas de un proyecto software.
2. Simulación de control de versiones de un proyecto open source.

Para acometer lo anterior el software desarrollado deberá implementar las siguientes funcionalidades:

1. Obtención de información de la gestión de tareas de un proyecto alojado en GitHub con la extensión ZenHub activada:
 - a) Sprints.
 - b) Tareas.
2. Obtención de información del control de versiones de un proyecto alojado en GitHub:
 - a) Commits.
3. Comparación de la información de la gestión de tareas de dos proyectos.
4. Comparación de la información del control de versiones de dos proyectos.
5. Aplicación de rúbrica de evaluación sobre:

- a) La información de gestión de tareas y del control de versiones de un repositorio.
- b) La comparación de la información de gestión de tareas y de control de versiones de dos repositorios, el original y el resultante de la simulación.

Los objetivos técnicos del desarrollo son:

1. Uso de técnicas de gestión de proyectos ágiles.
2. Cumplimiento de estándares de calidad de software y aseguramiento de calidad mediante análisis automatizados con Sonarqube.
3. Integración con GitHub.
4. Integración con ZenHub.
5. Uso de una arquitectura hexagonal [3] que desacople la funcionalidad desarrollada de los orígenes de datos. Esto facilitará la extensibilidad de la funcionalidad simplificando la integración con otros gestores de tareas y gestores de repositorios Git.

Conceptos teóricos

3.1. Conceptos de la gestión de proyectos ágiles utilizados en el laboratorio virtual.

Se presentan a continuación de forma breve conceptos de la gestión de proyectos ágiles utilizados en el laboratorio virtual, consultar [1] para obtener un conocimiento más detallado de los conceptos.

Historia de usuario.

Es una descripción informal de una funcionalidad del sistema a desarrollar. Se documentan desde la perspectiva del usuario final del sistema.

TBD - Incluir captura historia de usuario.

Puntos de historia de usuario.

Es una métrica utilizada por los miembros del equipo de proyecto para expresar el esfuerzo estimado requerido para completar una tarea. En la estimación se considera:

1. Complejidad de la tarea.
2. Cantidad de trabajo de la tarea.
3. Riesgos, incertidumbres de la tarea a realizar.

Sprint/Milestone/Hito.

Es un periodo de tiempo durante el cual se realiza un trabajo específico, definido en las distintas tareas que componen el sprint. En el laboratorio virtual es de relevancia la siguiente información:

TBD - Incluir captura sprint

1. Fecha de inicio.
2. Fecha de fin.
3. Puntos de historia de usuario: total de puntos de historia de usuario de las tareas incluidas en el sprint.

Tarea/Issue.

Cada historia de usuario es dividida en unidades de trabajo individuales. En el laboratorio virtual se plantea especificar la siguiente información:

1. Descripción de la tarea.
2. Comentarios de una tarea: Una tarea puede incluir comentarios, cada comentario tiene un autor.
3. Miembro del equipo al que se le asigna la tarea.
4. Puntos de historia de usuario.
5. Tipos de tarea:
 - a) Issue: tarea de propósito general.
 - b) Bug: incidencia.
6. Estados de una tarea:
 - a) Open/abierta.
 - b) Closed/cerrada.
7. Etiquetas: permiten la clasificación de una tarea en diversas categorías. Existen etiquetas predefinidas y etiquedadas personalizadas definidas por el usuario. A efectos de la simulación se consideran las siguientes etiquetas predefinidas:

- a) Feature/funcionalidad.
- b) Testing/pruebas.
- c) Documentation/documentación.

TBD - Incluir captura tarea

3.2. Conceptos de control de versiones utilizados en el laboratorio virtual

Respecto a el control de versiones, se explican a continuación los conceptos utilizados en este trabajo.

Control de versiones

Como indica [7], el control de versiones consiste en la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

En el contexto de las tecnologías de información, este control de versiones se realiza mediante herramientas que permiten gestionar activos de elementos software como:

- Ficheros de código fuente a partir de los que se generan programas informáticos.
- Ficheros de configuración.
- Otros tipos de ficheros con formato binario empleados en el desarrollo de programas informáticos, como por ejemplo imágenes.

En la actualidad el uso de herramientas de control de versiones se extiende fuera de la industria de software a otras industrias, como por ejemplo a la industria de la edición y publicación de libros.

De entre todas la herramientas de control de versiones, Git, es el estándar de facto.

Git

En este trabajo se utilizan los siguientes términos pertenecientes al glosario de términos [4] de Git.

- Repositorio: colección de enlaces a objetos y base de datos de objetos en los que se almacenan los elementos a gestionar.
- Rama/Branch: línea de desarrollo. Cada rama mantiene su propia colección de enlaces.
- Commit: Un punto individual de la historia de Git. Entre otros datos, contiene información de:
 - Autor.
 - Fecha y hora del commit.
 - Comentarios asociados.

GitHub

Es un servicio de alojamiento en internet de repositorios Git.

3.3. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.4. Referencias

Las referencias se incluyen

Herramientas	App AngularJS
HTML5 More H More	X

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

3.5. Imágenes

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

3.6. 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.

▪

3.7. Tablas

Igualmente se pueden usar los comandos específicos de L^AT_EXo bien usar alguno de los comandos de la plantilla.

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 3.2: Herramientas y tecnologías utilizadas en cada parte del proyecto

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.

4.1. Arquitectura hexagonal

TBD.

4.2. Integración continua

Pipeline

TBD.

Sonarqube

TBD.

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.

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.

Conclusiones y Líneas de trabajo futuras

TBD Conclusiones.

Se plantean las siguientes líneas de trabajo futuras:

1. Ampliar la información a evaluar en las simulaciones . Esto conllevará ampliar el modelo de dominio así como los datos obtenidos en las integraciones con GitHub y ZenHub:
 - a)* Wiki.
 - b)* Pull requests.
 - c)* Estimación de tipo póker.

Bibliografía

- [1] Agile Alliance. Agile glossary. [Internet; consultado 14-octubre-2022].
- [2] John Bourne, Dale Harris, and Frank Mayadas. Online engineering education: Learning anywhere, anytime. *Online Learning*, 9, 03 2019.
- [3] Alistair Cockburn. Hexagonal architecture, 2005. [Internet; consultado 14-octubre-2022].
- [4] Git. gitglossary - a git glossary, 2022. [Internet; descargado 25-octubre-2022].
- [5] Pilar Sánchez, Ángel Garnacho, Jacinto Dacosta, and Enrique Mandado. El aprendizaje activo mediante la autoevaluación utilizando un laboratorio virtual. *IEEE-RITA*, 4:53–62, 01 2009.
- [6] Inés Tejado, Emiliano Hernández, Isaías González, Pilar Merchán, and Blas Vinagre. Introducing automatic evaluation in virtual laboratories for control engineering at university of extremadura. first steps. *International Journal of Mathematics and Computers in Simulation*, 12:55–63, 04 2018.
- [7] Wikipedia. Control de versiones — wikipedia, la enciclopedia libre, 2022. [Internet; descargado 25-octubre-2022].