



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.  
Documentación Técnica**



Presentado por Licinio Fernández Maurelo  
en Universidad de Burgos — 22 de mayo  
de 2023

Tutor: Dr. Carlos López Nozal



---

# Índice general

---

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	5
<b>Apéndice B Especificación de Requisitos</b>	<b>9</b>
B.1. Introducción . . . . .	9
B.2. Objetivos generales . . . . .	9
B.3. Catalogo de requisitos . . . . .	9
B.4. Especificación de requisitos . . . . .	9
<b>Apéndice C Especificación de diseño</b>	<b>11</b>
C.1. Introducción . . . . .	11
C.2. Diseño de datos . . . . .	11
C.3. Diseño procedimental . . . . .	13
C.4. Diseño arquitectónico . . . . .	13
<b>Apéndice D Documentación técnica de programación</b>	<b>15</b>
D.1. Introducción . . . . .	15
D.2. Estructura de directorios . . . . .	15
D.3. Manual del programador . . . . .	15

D.4. Compilación, instalación y ejecución del proyecto . . . . .	15
D.5. Pruebas del sistema . . . . .	15
<b>Apéndice E Documentación de usuario</b>	<b>17</b>
E.1. Introducción . . . . .	17
E.2. Requisitos de usuarios . . . . .	17
E.3. Instalación . . . . .	17
E.4. Manual del usuario . . . . .	17
<b>Bibliografía</b>	<b>19</b>

---

# Índice de figuras

---

A.1. Java Developer Hourly Rate Guide . . . . .	6
C.1. Modelo de datos para la gestión de proyectos ágiles . . . . .	12
C.2. Modelo de datos para el control de versiones . . . . .	13

---

# Índice de tablas

---

B.1. CU-1 Nombre del caso de uso. . . . .	10
---	----

## *Apéndice A*

---

# **Plan de Proyecto Software**

---

### **A.1. Introducción**

En todo proyecto software es fundamental considerar los aspectos relativos a:

- Planificación temporal
- Estudio de viabilidad económica y legal

### **A.2. Planificación temporal**

La planificación temporal es un componente vital de la gestión de proyectos, que contribuye a la finalización exitosa del mismo. La gestión temporal eficaz de un proyecto implica:

- Creación de un cronograma de proyecto realista
- Identificación de hitos clave
- Definición de entregables

En este trabajo se ha hecho uso de metodología scrum, dividiendo las tareas del proyecto en sprints de duración de dos semanas como norma general si bien en algunos casos estos sprints se han ampliado en duración.

El detalle de tareas que compone cada sprint puede consultarse en **GitHub**<sup>1</sup> Se enumeran a continuación los sprints que se han realizado en el proyecto junto con una descripción breve de los mismos.

**Sprint 0 - Kick-off initial project documentation.** En este primer sprint

- revisamos la temática de este trabajo y comenzamos la redacción de los apartados de introducción y objetivos.
- se creó el repositorio GitHub como soporte de los activos documentales.
- se realizaron unas primeras pruebas de trabajo con las plantillas L<sup>A</sup>T<sub>E</sub>X correspondiente a la memoria y anexos de este trabajo.

**Sprint 1 - Initial domain model, GitHub integration .** En este segundo sprint:

- se creó el proyecto Java asociado.
- se codificaron los primeros elementos del modelo.
- experimentamos con el API REST de GitHub y realizamos pruebas creando un cliente del API REST personalizado utilizando Open Feign.
- se configuró la integración continua para poder tener un feedback actualizado de métricas fundamentales en la calidad de software (cobertura de tests, duplicados, code smells).

**Sprint 2 - Import agile project management data.** En este sprint:

- se implementa la ingesta de información de la gestión ágil de proyectos.
- se incluyen tests funcionales utilizando Cucumber.
- se analiza la problemática de la integración de la información de los puntos de historia de usuario desde ZenHub.

**Sprint 3 - Import issue events detail.** En este sprint:

---

<sup>1</sup>GitHub, sprints finalizados: <https://github.com/lfx1001/avrela/milestones?state=closed>



- se complementó la información relativa a la gestión de proyectos ágiles añadiendo e integrando la parte de eventos de las tareas.
- se llevaron a cabo tareas de documentación de algunas de las herramientas y técnicas utilizadas hasta el momento.

**Sprint 4 - Rubric evaluation (Teamwork).** En este sprint:

- se trabaja el aspecto de evaluación de las simulaciones.
- se implementa la evaluación de trabajo el equipo en la gestión de proyectos ágiles utilizando Cucumber.

**Sprint 5 - Initial similarity function implementation.** En este sprint:

- se implementa la función de semejanza para la comparación de tareas utilizando el índice de Jaccard.
- se evalúa el criterio de semejanza de descripción de tareas.

**Sprint 6 - SCM evaluation.** En este sprint:

- se implementa la importación de commits dado un intervalo de tiempo.
- se implementa la función de semejanza de commits. Se utiliza la distancia de Jaro-Wincker junto con el índice de Jaccard.
- se evalúa el criterio de trazabilidad de la simulación de control de versiones con la herramienta de gestión de proyectos.
- se evalúa el criterio de trabajo en equipo de la simulación de control de versiones.
- se evalúa el criterio de semejanza de la simulación de control de versiones.

**Sprint 7 - Agile project management case study simulation UI.** En este sprint:

- se realizan pruebas de implementación de interfaces de usuario con Thymeleaf.
- se implementa una versión inicial de pantallas para la evaluación de proyectos ágiles.

**Sprint 8 - Integrate APM SCM case study simulation backend with the UI.** En este sprint:

- se implementa una versión inicial de pantallas para la evaluación de control de versiones.
- se integran las pantallas de evaluación de gestión ágil de proyectos con la lógica de negocio.
- se integran las pantallas de evaluación de control de versiones con la lógica de negocio.

**Sprint 9 - Documentation Web application improvements.** En este sprint:

- se corrigen incidencias de las evaluación.
- se realizan mejoras de la interfaz visual de la evaluación de simulaciones de control de versiones.
- se trabaja en la documentación de anexos y memoria.

**Sprint 10 - Documentation Web application improvements.** En este sprint:

- se corrigen incidencias de las evaluación.
- se realizan mejoras de la interfaz visual de la evaluación de simulaciones de control de versiones.
- se trabaja en la documentación de anexos y memoria.

## A.3. Estudio de viabilidad

### Viabilidad económica

Los costes de este proyecto software se basan en las siguientes líneas:

- Coste de desarrollo.
- Coste de recursos asociados.

En este caso vamos a optar por un cálculo simplificado que incluya los propios recursos asociados en el coste de desarrollo bajo el supuesto de que este proyecto se gestionase "llave en mano". De esta forma, el cliente (en esta caso, la universidad) encargaría a un tercero (el alumno) la realización del proyecto como un proveedor externo.

Este proveedor externo gestionaría internamente los gastos correspondientes recursos humanos y materiales. Proporcionaría al cliente una cifra final de costes. Este coste está determinado por la expresión:

$$\text{Coste} = \text{Jornadas} \times \text{Tarifa jornada}$$

Jornadas representa el número de jornadas necesarias para acometer el proyecto. Se estima en 40 jornadas (320 horas).

Tarifa jornada representa el coste unitario de la jornada. Este coste está relacionado con el perfil del recurso que vaya a acometer el proyecto. Para el cálculo se considera un perfil de desarrollador Java Senior. La tarifa presenta una gran variabilidad en función de la localización. Aprovecharemos el estudio realizado por **flexiple**<sup>2</sup> cuya captura se recoge en A.1. Este estudio estima en 55,43 Euros (60 Dólares) la tarifa hora, lo que supone un coste de 443,7 Euros por jornada.

---

<sup>2</sup><https://flexiple.com/java/hourly-rate/>

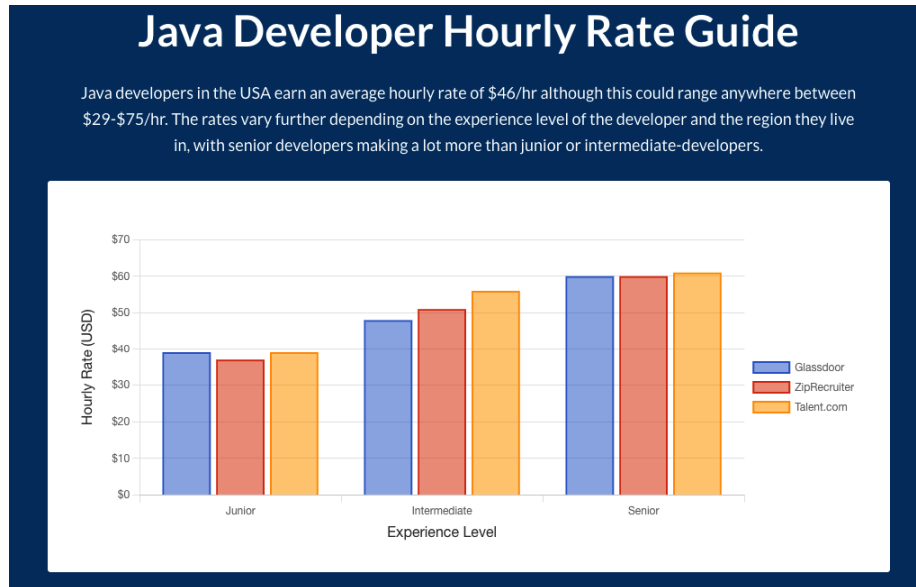


Figura A.1: Java Developer Hourly Rate Guide

En base a los supuestos anteriores, una estimación posible del coste del proyecto, impuestos no incluidos, es:

$$40 \text{ Jornadas} \times 443,7 \text{ Euros Tarifa jornada} = 17.748 \text{ Euros}$$

Las funcionalidades proporcionados por este trabajo permiten automatizar tareas de evaluación en el ámbito académico. Para optimizar el retorno de inversión estas funcionalidades podrían ampliarse y explotarse en otros ámbitos. Se proponen a continuación dos supuestos en los que la herramienta podría ser de interés comercial:

- Evaluación de competencias técnicas en procesos de selección personal: los candidatos podrían realizar una simulación que sería contrastada con el caso de estudio de referencia con el fin de determinar si son aptos o no en la competencia en cuestión.
- Generación de planificaciones a partir de casos de estudio: la herramienta podría ingestar una planificación de proyectos ágiles de un sistema origen y copiarla a un sistema destino. La variedad de adaptadores de la herramienta determinarían el grado de interoperabilidad. Actualmente, la herramienta únicamente permite extraer información de

GitHub por lo que sería necesario implementar adaptadores para otras herramientas, como **JIRA**<sup>3</sup>, tanto de entrada como de salida.

## Viabilidad legal

La viabilidad legal abarca el cumplimiento de las leyes, reglamentos y derechos de propiedad intelectual pertinentes. Implica consideraciones tales como licencias, privacidad de datos, seguridad y cualquier obligación legal asociada con el proyecto de software. Al abordar de manera proactiva los aspectos legales, mitigamos los riesgos potenciales y nos aseguramos de que el proyecto opere dentro de los límites de las leyes aplicables.

Se listan a continuación las librerías y agrupaciones de librerías (indicadas con la terminación `.*`) con su licencia correspondiente:

- **org.springframework.boot.\*** : licencia Apache-2.0
- **org.springframework.cloud.\*** : licencia Apache-2.0
- **org.webjars:bootstrap** : licencia MIT
- **org.junit.\*** : Licencia Pública Eclipse - v 2.0
- **io.cucumber** : licencia MIT

Las licencias anteriores nos permitirían el uso de las librerías con fines comerciales.

---

<sup>3</sup>JIRA: <https://www.atlassian.com/software/jira>



## *Apéndice B*

---

# **Especificación de Requisitos**

---

### **B.1. Introducción**

Una muestra de cómo podría ser una tabla de casos de uso:

### **B.2. Objetivos generales**

### **B.3. Catalogo de requisitos**

### **B.4. Especificación de requisitos**

CU-1	Ejemplo de caso de uso
<b>Versión</b>	1.0
<b>Autor</b>	Alumno
<b>Requisitos asociados</b>	RF-xx, RF-xx
<b>Descripción</b>	La descripción del CU
<b>Precondición</b>	Precondiciones (podría haber más de una)
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pasos del CU</li> <li>2. Pasos del CU (añadir tantos como sean necesarios)</li> </ol>
<b>Postcondición</b>	Postcondiciones (podría haber más de una)
<b>Excepciones</b>	Excepciones
<b>Importancia</b>	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.



## *Apéndice C*

---

# **Especificación de diseño**

---

### **C.1. Introducción**

### **C.2. Diseño de datos**

En este proyecto se trabaja con varios dominios de datos:

- Gestión de proyectos ágiles.
- Control de versiones.

El dominio de la gestión de proyectos ágiles se refleja en el diagrama de clases [C.1](#).



Figura C.1: Modelo de datos para la gestión de proyectos ágiles

El dominio para modelar el control de versiones obedece al diagrama de clases [C.2](#).

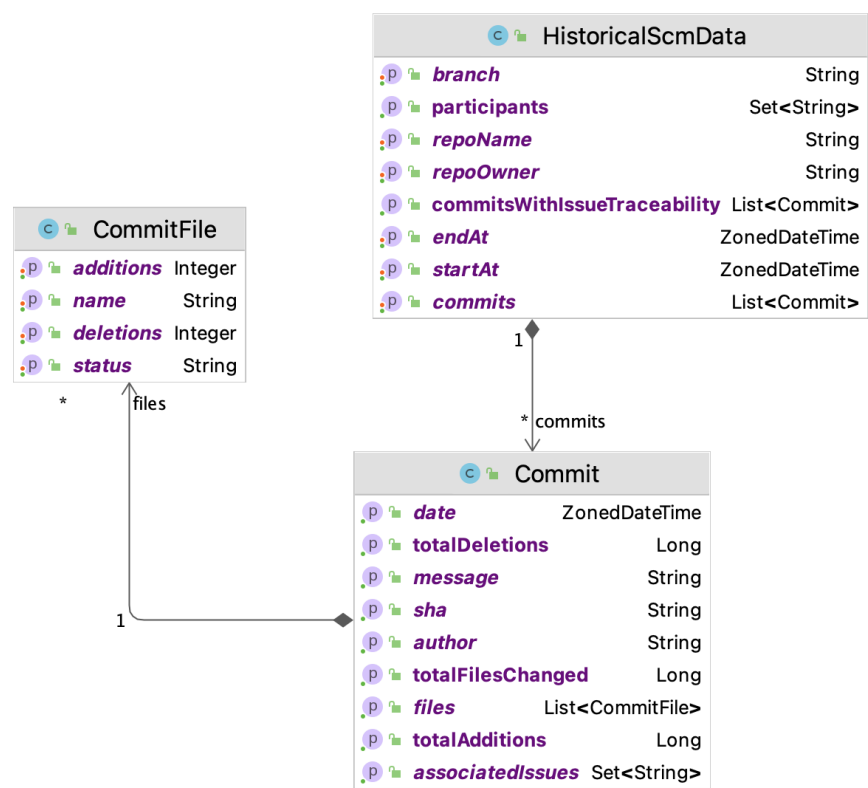


Figura C.2: Modelo de datos para el control de versiones

C.3. Diseño procedimental

C.4. Diseño arquitectónico



## *Apéndice D*

---

# **Documentación técnica de programación**

---

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema



## *Apéndice E*

---

# **Documentación de usuario**

---

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario





---

## **Bibliografía**

---