



ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENIERIA EN SISTEMAS
AUDIOVISUALES Y MULTIMEDIA

TRABAJO FIN DE GRADO

TÍTULO DEL TRABAJO EN MAYÚSCULAS

Autor : Paula Sepúlveda Membrilla

Tutor : Gregorio Robles Martínez

Curso académico 2023/2024

Trabajo Fin de Grado

Título del Trabajo con Letras Capitales para Sustantivos y Adjetivos

Autor : Paula Sepúlveda Membrilla

Tutor : Gregorio Robles Martínez

La defensa del presente Proyecto Fin de Carrera se realizó el día de
de 202X, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 202X

*Dedicado a
mi familia / mi abuelo / mi abuela*

Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.

Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Definición del problema	2
1.3. Objetivos e hipótesis	2
1.4. Estructura de la memoria	3
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	6
2.3. Planificación temporal	7
3. Estado del arte	9
3.1. Tecnologías y herramientas	9
3.1.1. Python	9
3.1.2. SQL	9
3.2. Librerías	9
4. Diseño e implementación	11
4.1. Arquitectura general	11
5. Experimentos y validación	13
6. Resultados	15
7. Conclusiones	17
7.1. Consecución de objetivos	17

7.2. Aplicación de lo aprendido	17
7.3. Lecciones aprendidas	18
7.4. Trabajos futuros	18
A. Manual de usuario	19

Índice de figuras

2.1. Diagrama de Gantt	7
4.1. Estructura del parser básico	12

Capítulo 1

Introducción

En este capítulo se presenta la motivación que ha impulsado esta investigación, proporcionando el contexto y las razones fundamentales para llevarla a cabo. A continuación, se define claramente el problema a resolver, especificando su alcance y relevancia, así como los objetivos que se desean alcanzar y las hipótesis planteadas al inicio del estudio.

Finalmente, se describe una visión general de cómo está organizado el documento y qué secciones lo componen.

1.1. Motivación

La inspiración que me ha llevado a realizar este proyecto proviene de mi profesor y tutor, Gregorio Robles, quién me ha motivado a adentrarme en el mundo del software libre (OSS, del inglés *Open Source Software*).

El término software libre hace referencia al código diseñado de manera descentralizada y colaborativa, siendo accesible para todo el público. Esto permite que cualquiera pueda utilizarlo, examinarlo, modificarlo y redistribuirlo como considere conveniente.

La elección de código abierto nos aporta beneficios tales como un bajo coste, gran flexibilidad para modificar el código fuente, así como un buen soporte proporcionado por parte de la comunidad. Sin embargo, no existe una clara distinción entre el proceso de desarrollo y el de mantenimiento, lo cual resulta interesante para la Ingeniería de Software al estudiar y explicar el funcionamiento de las interacciones que se llevan a cabo abiertamente.

Por lo tanto, la principal motivación de esta investigación es analizar el comportamiento de

los proyectos de código abierto, los cuales se desarrollan en una comunidad de usuarios que se comunican a través de distintas herramientas, teniendo como factor común Internet, lo que permite guardar el registro de las actividades realizadas a lo largo del tiempo.

Además, este proyecto me ha brindado la oportunidad de adquirir conocimientos sobre minería de datos y desarrollar habilidades para realizar un correcto análisis de los datos obtenidos.

1.2. Definición del problema

El software libre permite a los desarrolladores realizar un completo análisis cuantitativo del código y de todos los parámetros involucrados en su producción, debido a que están disponibles públicamente. Al contar con datos de acceso público del desarrollo de software, permite llevar a cabo estudios estadísticos para estudiar la evolución del software.

En todo este proceso, tiene gran importancia la utilización de un sistema de control de versiones, donde se permita rastrear los cambios realizados en el pasado. De esta manera, podemos obtener un análisis de la estructura histórica del código en base a sus distintas contribuciones, así como harían los arqueólogos estudiando una ciudad en base a sus distintas construcciones. Para software libre esto tiene vital importancia, ya que debido a la falta de diseño, el mantenimiento del código es una actividad relevante para evitar posible código obsoleto de desarrolladores que ya no participan en el proyecto.

Aunque la Ingeniería de Software es una disciplina que se ha consolidado a través de los años, actualmente existen muy pocos análisis empíricos sobre la arqueología del software. La falta de estudios junto con el atractivo hacia el desarrollo comunitario del software libre, así como la importancia para la Ingeniería de Software de modelar con datos empíricos, nos impulsa a llevar a cabo este proyecto.

1.3. Objetivos e hipótesis

Nuestra hipótesis es que, mediante un estudio empírico de las prácticas de desarrollo de software, la aplicación de minería de datos al proyecto, el análisis de su evolución a lo largo del tiempo y la correlación de datos, se pueden determinar y aplicar métricas que demuestren que la longevidad y el mantenimiento del código son heterogéneos y varían entre los diferentes

componentes de varios proyectos.

Por lo que, el objetivo del proyecto consiste en estudiar la evolución y propiedades de varios proyectos de software libre a través del estudio empírico, y en aplicar un análisis estadístico de su código fuente. Se estudia el comportamiento de los proyectos FOSS¹ más relevantes en GitHub, se identifican las fuentes de datos que ofrecen de manera pública y se presenta una metodología para el análisis de los datos extraídos, que nos aportan datos interesantes a la investigación basados en la cantidad de colaboradores o en los lenguajes de programación más utilizados. De esta manera, podemos conocer mejor el fenómeno del software libre, el proceso de creación de software y cómo se puede aplicar en cualquier otro entorno de desarrollo, se describe la mecánica de desarrollo de los movimientos de software libre.

Tenemos el objetivo de comprender el desarrollo de proyectos de software libre, así como la posibilidad de renovar la mecánica llevada a cabo mediante la aplicación de los resultados obtenidos en esta investigación.

1.4. Estructura de la memoria

A continuacion, describimos la estructura de la memoria, exponiendo el contenido de cada uno de los capítulos, proporcionando así una guía organizada del trabajo de fin de grado para una mejor lectura y comprension de este.

■ Capítulo 1: Introducción.

En este capítulo se describe la motivación que me ha llevado a realizar la investigación, así como el contexto, la definición del problema y las distintas hipótesis que se han desarrollado. Además, se ha ofrecido una breve descripción de la estructura del proyecto.

■ Capítulo 2: Objetivos.

En este capítulo se describe el objetivo principal de la investigación, así como los objetivos específicos necesarios que van a guiar nuestro proyecto. Incluye una planificacion temporal de los objetivos mencionados.

■ Capítulo 3: Estado del arte.

En este capítulo se proporciona información detallada sobre el diseño, las características

¹Free/Open Source Software

y los usos de cada una de las tecnologías y herramientas utilizadas en el proyecto.

■ **Capítulo 4: Diseño e implementación.**

En este capítulo se detallan las fases que se han llevado a cabo para realizar el análisis de estudio. Se describe la arquitectura general del proyecto, la mecánica de obtención y almacenamiento de los datos, y su posterior análisis empírico.

■ **Capítulo 6: Resultados.**

En este capítulo se describen los resultados obtenidos tras analizar los datos recopilados en el análisis, y se realiza una justificación de dichos resultados.

■ **Capítulo 7: Conclusiones.**

En este capítulo se explican las conclusiones a las que se ha llegado, con su correspondiente justificación. Además, se mencionan los conocimientos adquiridos durante la duración de todo el grado universitario que se han aplicado en este proyecto, y los relacionados con el aprendizaje adquirido durante la realización de este estudio. Por último, se mencionan posibles mejoras que se pueden aplicar en el futuro.

Capítulo 2

Objetivos

En este capítulo, se definen los objetivos que guiarán la realización de este Trabajo de Fin de Grado (TFG). Establecer objetivos claros y bien definidos es fundamental para asegurar una dirección precisa y coherente a lo largo del desarrollo del proyecto.

Estos objetivos se dividen en dos categorías principales: el objetivo general, que representa la meta global del trabajo, y los objetivos específicos, que son metas más concretas y detalladas necesarias para alcanzar el objetivo general.

Además, se presenta una planificación temporal que muestra la organización de las tareas a lo largo del período de ejecución del proyecto, garantizando un avance ordenado y sistemático.

2.1. Objetivo general

El objetivo general del proyecto es investigar en profundidad la evolución y las características de proyectos de software libre alojados en la plataforma de GitHub mediante un estudio empírico detallado y la aplicación estadística de su código fuente. A través de este enfoque, se pretende obtener una comprensión íntegra de cómo el proyecto ha cambiado y se ha desarrollado a lo largo del tiempo, identificando patrones, tendencias y características del código que puedan influir en su calidad, mantenibilidad y eficiencia.

Esta investigación no solo implica un análisis histórico, sino que también incluye la aplicación de diversas métricas para evaluar distintos aspectos del código fuente. Al combinar métodos empíricos con análisis estadísticos robustos, este proyecto busca proporcionar una visión detallada y objetiva del comportamiento evolutivo del software.

Los resultados obtenidos nos ofrecen conclusiones y recomendaciones para la mejora continua del desarrollo del software libre, contribuyendo al conocimiento en el campo de Ingeniería de Software que puede aplicarse en la toma de decisiones en proyectos futuros.

2.2. Objetivos específicos

Para lograr el objetivo general detallado en el proyecto se han establecido los siguientes objetivos específicos:

- **Investigación.**

Consiste en buscar y evaluar información sobre el contexto de trabajo, realizando una estimación de los costes de desarrollo software. Se piensa como desarrollar la programación de manera eficiente, y a su vez se realiza una selección de distintos repositorios de GitHub que resultan de especial interés para nuestra investigación.

- **Programación.**

En esta fase del proyecto se desarrollan los scripts necesarios para extraer, almacenar y analizar los datos con el fin de conseguir los objetivos de esta investigación.

- **Recopilación de datos.**

En esta etapa se recopilan y almacenan los datos más relevantes para el posterior estudio del comportamiento de los repositorios seleccionados.

- **Análisis estadístico.**

Tras la recopilación de los datos, se ajustan y configuran las métricas correspondientes para obtener distintas estadísticas sobre la evolución de los repositorios a lo largo del tiempo.

- **Representación gráfica.**

Una vez que se ha realizado un análisis estadístico, se realiza un análisis empírico y se representan de manera gráfica los resultados obtenidos.

- **Documentación.**

En esta etapa se elaboran la documentación y los informes necesarios para la correcta entrega del Trabajo de Fin de Grado (TFG).

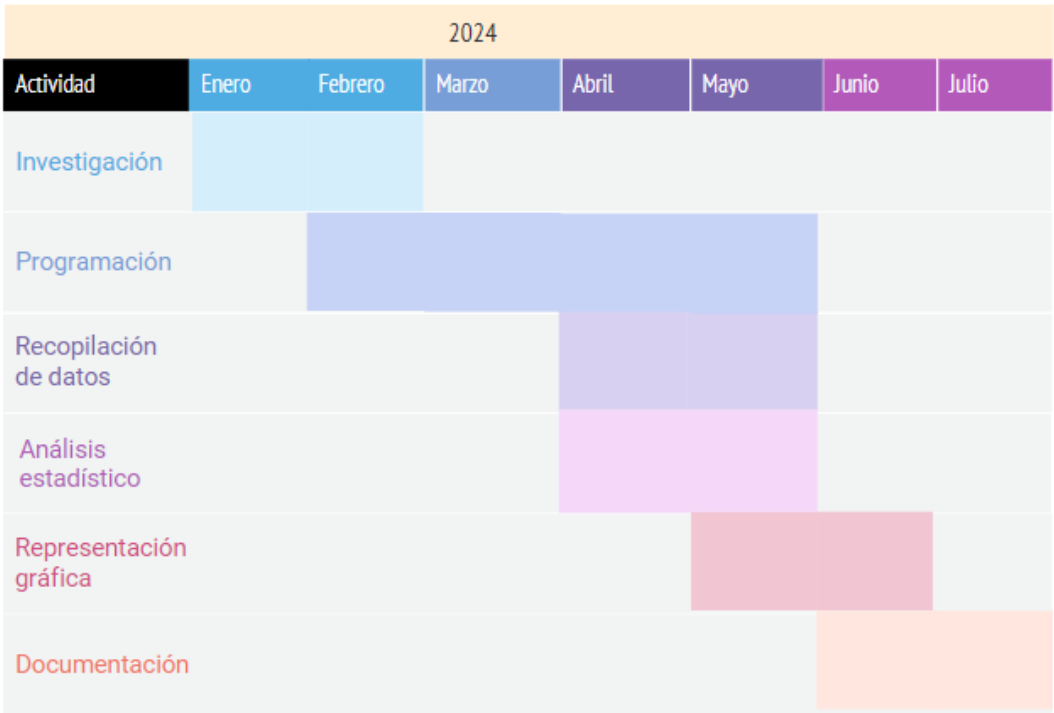


Figura 2.1: Diagrama de Gantt

2.3. Planificación temporal

El desarrollo de este proyecto abarca un período de siete meses, desde enero de 2024 hasta julio de 2024, con la realización de un trabajo diario que se lleva a cabo de manera constante. En la figura 2.1 se observa el tiempo empleado en las diferentes fases del proyecto, donde se puede observar que varias tareas se han solapado en algunos casos debido al avance en paralelo de distintas actividades.

■ **Enero 2024**

En este mes comienzo el desarrollo de esta investigación, cuando mi tutor Gregorio Robles me sugirió la idea del proyecto. Comienzan las reuniones donde se definen los objetivos correspondientes, y empiezo a documentarme para obtener los conocimientos necesarios para la realización del proyecto.

Recopilo información acerca de los proyectos de código abierto y posibles repositorios de especial interés que son alojados en la plataforma de GitHub. A su vez, estudio la información que se puede obtener de las líneas de código mediante el comando *git blame*, y que análisis nos puede resultar de interés para estudiar la evolución de estos repositorios

a lo largo de los años.

■ Febrero 2024

Durante este mes sigo realizando la tarea de investigación, pero además trabajo en paralelo con la tarea de programación. Comienzo a desarrollar el código de programación basado en el comando *git blame*, mediante el cual obtengo información interesante acerca de los cambios realizados en un archivo específico línea por línea. El comando *git blame* nos proporciona información del autor de la línea de código, así como de la fecha del commit, del estado de esa línea y obviamente podemos ver también su contenido.

■ Marzo 2024

En este mes me centro exclusivamente en la programación del código, desarrollando los scripts *init.py* y *git-blameall.py*. Estos scripts analizan un repositorio y un archivo respectivamente, con el objetivo de analizar línea a línea cada uno de los archivos de un repositorio, y guardan los datos recogidos en un archivo JSON.

Al principio, se trabaja en realizar el análisis para un único archivo y finalmente se logra conseguir analizar un repositorio concreto. Además, se estudia la opción de tener que analizar las distintas versiones de un archivo o unicamente la versión final, quedandonos con la segunda opción al ser mucho más eficiente y conseguir obtener los datos que nos interesan.

■ Abril 2024

Este mes tiene una gran carga de trabajo, al realizar en paralelo tres actividades distintas. Por una parte, sigo avanzando con la programación, mejorando los scripts *init.py* y *git-blameall.py* de manera que los datos pasan de guardarse en un archivo JSON a guardarse en tres bases de datos relacionadas entre sí: *repositories*, *files* y *code* respectivamente. Esto requiere de un estudio previo para obtener los conocimientos necesarios sobre la utilización de *MySQL* y el concepto de bases de datos relacionales. El hecho de almacenar los datos en bases de datos relacionales nos permite un posterior análisis de los datos de manera mucho más eficiente.

■ Mayo 2024

Durante estas semanas la carga de trabajo sigue siendo muy alta, desarrollando cuatro

actividades de manera paralela. Termino de desarrollar los archivos de programación, incluido entre ellos el script *graphics.py* que tiene como finalidad poder obtener una representación gráfica de las métricas estudiadas. A su vez, termino de definir los índices y las métricas que se incluyen en este proyecto, para obtener estadísticas interesantes acerca de la evolución de cambios, número de autores o posible código obsoleto en un repositorio. Seguidamente, se recopilan los últimos datos de los repositorios elegidos para su análisis, todos ellos alojados en la plataforma de GitHub.

- **Junio 2024**

En esta etapa se elaboran las gráficas definitivas, y mientras tanto se empieza a escribir la memoria de nuestro proyecto.

- **Julio 2024**

En este último mes se termina de escribir la memoria y se realizan las correcciones oportunas por el tutor para poder realizar una correcta entrega del Trabajo de Fin de Grado (TFG).

Capítulo 3

Estado del arte

En este capítulo, se presentan las herramientas y librerías usadas en el Trabajo Fin de Grado. Esta exposición nos da una visión de las tecnologías empleadas en el proyecto.

3.1. Tecnologías y herramientas

Hemos hablado de cómo incluir figuras. Pero no hemos dicho nada de tablas. A mí me gustan las tablas. Mucho. Aquí un ejemplo de tabla, la Tabla 3.1 (siento ser pesado, pero nota cómo he puesto la referencia).

3.1.1. Python

—

3.1.2. SQL

3.2. Librerías

Hay un sitio en Internet donde puedes diseñar las tablas fácilmente y luego hacer un corta y pega del resultado en tu editor. Puedes probarlo en <https://www.tablesgenerator.com/>.

Uno	2	3
Cuatro	5	6
Siete	8	9

Cuadro 3.1: Ejemplo de tabla. Aquí viene una pequeña descripción (el *caption*, el pie de tabla/figura) del contenido de la tabla. Si la tabla no es autoexplicativa, siempre viene bien aclararla aquí.

Capítulo 4

Diseño e implementación

Aquí viene todo lo que has hecho tú (tecnológicamente). Puedes entrar hasta el detalle. Es la parte más importante de la memoria, porque describe lo que has hecho tú. Eso sí, normalmente aconsejo no poner código, sino diagramas.

4.1. Arquitectura general

Si tu proyecto es un software, siempre es bueno poner la arquitectura (que es cómo se estructura tu programa a “vista de pájaro”).

Por ejemplo, puedes verlo en la figura 4.1. \LaTeX pone las figuras donde mejor cuadran. Y eso quiere decir que quizás no lo haga donde lo hemos puesto... Eso no es malo. A veces queda un poco raro, pero es la filosofía de \LaTeX : tú al contenido, que yo me encargo de la maquetación.

Recuerda que toda figura que añadas a tu memoria debe ser explicada. Sí, aunque te parezca evidente lo que se ve en la figura 4.1, la figura en sí solamente es un apoyo a tu texto. Así que explica lo que se ve en la figura, haciendo referencia a la misma tal y como ves aquí. Por ejemplo: En la figura 4.1 se puede ver que la estructura del *parser* básico, que consta de seis componentes diferentes: los datos se obtienen de la red, y según el tipo de dato, se pasará a un *parser* específico y bla, bla, bla. . .

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación.

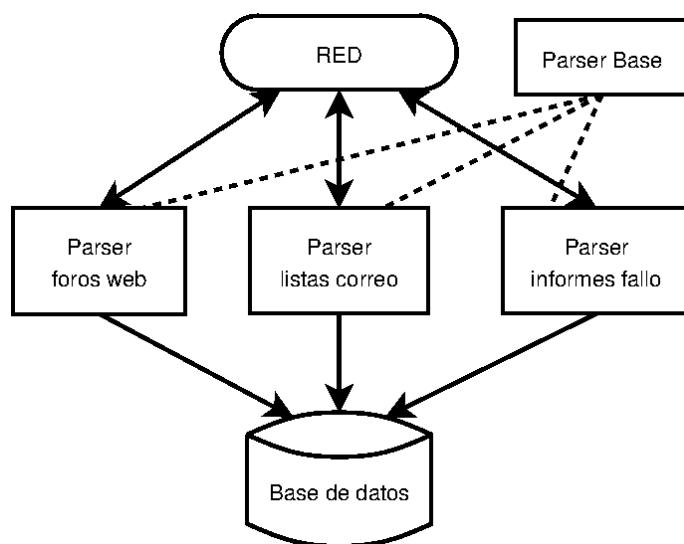


Figura 4.1: Estructura del parser básico

Capítulo 5

Experimentos y validación

Este capítulo se introdujo como requisito en 2019. Describe los experimentos y casos de test que tuviste que implementar para validar tus resultados. Incluye también los resultados de validación que permiten afirmar que tus resultados son correctos.

Capítulo 6

Resultados

En este capítulo se incluyen los resultados de tu trabajo fin de grado.

Si es una herramienta de análisis lo que has realizado, aquí puedes poner ejemplos de haberla utilizado para que se vea su utilidad.

Capítulo 7

Conclusiones

7.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

Y si has llegado hasta aquí, siempre es bueno pasarle el corrector ortográfico, que las erratas quedan fatal en la memoria final. Para eso, en Linux tenemos *aspell*, que se ejecuta de la siguiente manera desde la línea de *shell*:

```
aspell --lang=es_ES -c memoria.tex
```

7.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

7.3. Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. Aquí viene uno.
2. Aquí viene otro.

7.4. Trabajos futuros

Ningún proyecto ni software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFM.

Apéndice A

Manual de usuario

Esto es un apéndice. Si has creado una aplicación, siempre viene bien tener un manual de usuario. Pues ponlo aquí.

