



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFM del Máster en Ingeniería
Informática

Análisis Visual de Revisiones de
Código



Presentado por Mario Juez Gil
en Universidad de Burgos — 8 de marzo de 2017
Tutor: Carlos López Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



D. Carlos López Nozal, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Mario Juez Gil, con DNI 71308224J, ha realizado el Trabajo final de Máster en Ingeniería Informática titulado Análisis Visual de Revisiones de Código.

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 marzo de 2017

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Carlos López Nozal

D. Raúl Marticorena Sánchez

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	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	2
Conceptos teóricos	3
3.1. Revisión de código	3
3.2. Revisor de código	4
3.3. Control de versiones	4
3.4. Integración continua	4
Técnicas y herramientas	5
Aspectos relevantes del desarrollo del proyecto	6
Trabajos relacionados	7
Conclusiones y Líneas de trabajo futuras	8
Bibliografía	9

Índice de figuras

Índice de tablas

Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

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.

Conceptos teóricos

En este apartado se exponen una serie de conceptos teóricos relacionados con las revisiones de código. Se define cada concepto así como su relación con el presente trabajo.

3.1. Revisión de código

La revisión de código es un proceso de ingeniería a través del cual se realiza una inspección del código fuente por otros desarrolladores diferentes al autor del mismo. Resulta útil para reducir los defectos de código así como mejorar la calidad del software [1].

Frente a las revisiones de código altamente estructuradas propuestas por Fagan [3], hoy en día se están adoptando metodologías más livianas con el fin de solventar las ineficiencias de las inspecciones de código. Las denominadas *Modern Code Reviews* son informales, hacen uso de herramientas, y se utilizan regularmente.

Mediante el uso de estas nuevas metodologías, las revisiones de código ofrecen un mayor número de beneficios a los equipos de desarrollo como transferencia de conocimientos, visión de equipo, o mejores soluciones a los problemas [2].

Las revisiones de código son el elemento principal de este trabajo.

Inspecciones de código

Revisión automática de código

3.2. Revisor de código

Tipos de revisores

3.3. Control de versiones

Git

Repositorio

Cambio (diff)

Rama (branch)

Integración (merge)

Revision

Version (blob)

Árbol (tree)

Cambio (commit)

Etiqueta (tag)

Pull request

3.4. Integración continua

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.

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

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] A. F. Ackerman, L. S. Buchwald, and F. H. Lewski. Software inspections: an effective verification process. *IEEE Software*, 6(3):31–36, May 1989.
- [2] Alberto Bacchelli and Christian Bird. Expectations, outcomes, and challenges of modern code review. In *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, pages 712–721, Piscataway, NJ, USA, 2013. IEEE Press.
- [3] M. E. Fagan. Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 15(3):182–211, 1976.