

# Informe Seguimiento 2

## Aplicación de Gestión Financiera

Rafael Gómez Pérez  
Mención en Ingeniería del Software



## Índice de contenidos

1.	Histórico de Revisiones	3
2.	Objetivos del Proyecto_	4
3.	Planificación del Proyecto	4
4.	Metodología empleada	5
5.	Implementación de microservicios	6
6.	Presentación de resultados	6
7.	Conclusiones provisionales	6
8.	Listado de acrónimos	7
9.	Bibliografía y Referencias	7
10.	Anexo	8

# 1.Histórico de revisiones

Fecha	Versión	Descripción	Autor
22/05/2019	1.0	Creación del informe de progreso 2	Rafa

## 2 Objetivos del proyecto

Dado que nos encontramos ante el segundo informe de progreso, hemos creído oportuno mostrar de una forma algo mas visual la madurez de los objetivos establecidos al principio del proyecto.

La tabla a continuación muestra los códigos de los objetivos definidos previamente junto con un porcentaje aproximado que representa cuanto trabajo representado por cada objetivo esta actualmente completado.

Objetivo	Grado de Madurez
<b>tfg-obj-01</b>	80%
<b>tfg-obj-02</b>	100%
<b>tfg-obj-03</b>	50%
<b>tfg-obj-04</b>	40%
<b>tfg-obj-05.1</b>	45%
<b>tfg-obj-06</b>	70%
<b>tfg-obj-07</b>	0%

## 3 Planificación del Proyecto

En el apartado de diseño y arquitectura cubriremos las diferentes fases del proceso de diseño que se han realizado durante el período de tiempo entre este informe y el anterior.

En cuanto a la planificación del proyecto, actualmente nos encontramos entre 2 y 5 días por detrás de lo previsto inicialmente.

Esto se debe a la dificultad y la falta de experiencia en tecnologías como Docker Swarm, Spring Cloud (Netflix) y Spring Webflux. La falta de experiencia nos ha llevado a tener que realizar una cantidad de investigación mucho mayor a la esperada inicialmente, lo cual nos ha llevado a ralentizar el ritmo de trabajo.

Así mismo, la falta de madurez e información online con algunas de estas tecnologías Spring Webflux y Project Reactor (ambas lanzadas a principios de 2018) nos ha complicado y ralentizado el proceso de desarrollo.

## 4 Metodología de trabajo

En cuanto a metodología de trabajo, seguimos utilizando la misma que se comentó durante los informes previos (Kanban con feature-oriented). Pero, a diferencia de lo que se comentó en los informes mencionados se ha decidido realizar unos cambios en la parte de feature-oriented.

Anteriormente comentamos que nuestra implementación de feature-oriented sería la siguiente:

- Dividir el proyecto en funcionalidades claramente definidas.
- Separar y estimar cada funcionalidad según su impacto en cada lado de la aplicación Frontend o Backend.
- Agregar todas las tareas al backlog de Kanban.
- Escoger una funcionalidad e implementarla en ambos lados del proyecto antes de pasar a la siguiente.

Tras ver el tiempo restante y la dificultad que nos está llevando desarrollar la parte de backend, hemos decidido que cambiar nuestra implementación de feature-oriented a la siguiente:

- Dividir el proyecto en funcionalidades claramente definidas.
- Separar y estimar cada funcionalidad según su impacto en cada lado de la aplicación Frontend o Backend.
- Agregar todas las tareas al backlog de Kanban.
- Escoger un lado del proyecto (Front o Back) y desarrollar todas sus funcionalidades (Priorizamos Backend ya que existe una dependencia desde Front que necesita tener Back hecho).

Como podemos observar, la diferencia no mucha pues seguimos implementando feature-oriented programming simplemente modificamos como escoger funcionalidades a implementar.

## **5 Implementación de microservicios**

Como pudimos observar en el informe anterior, el proyecto consta de varios microservicios (consultar anexo 2ª imagen).

Respecto a su implementación, originalmente comentamos que el plan consistía en desarrollar la parte de Backend de forma monolítica para después de implementar el Frontend, realizar una migración hacía microservicios.

Tras pensar en otras alternativas, hemos decidido que creemos que es la mejor forma de hacerlo. Dada la casi inexistente experiencia laboral en entornos con arquitecturas basadas en microservicios con la que contamos, cabe la posibilidad que no hayamos podido realizar toda la migración de los microservicios antes de la fecha final del proyecto.

Teniendo esto en cuenta, creemos que el poder presentar un proyecto implementado de forma monolítica (en el peor de los casos) o híbrida en caso de no poder terminar la migración completa, nos beneficiará mucho mas que el poder mostrar todos los microservicios implementados y una aplicación (producto final) a medio implementar o con funcionalidad reducida significativamente.

## **6 Exposición de resultados**

## **7 Conclusiones provisionales**

Como conclusiones del proyecto (por el momento), podemos observar que la dificultad del proyecto y las tecnologías que están siendo utilizadas pueden haber sido algo mayores de lo que nos parecía inicialmente antes de ponernos a trabajar con ellas.

Creemos que la dificultad del proyecto no solo radica en las tecnologías con las que será implementado, sino en su propia lógica de negocio a pesar de que pueda parecer fácil a simple vista. Hasta el punto de haber tenido que incrementar la dificultad de este apartado para conseguir la aprobación del proyecto, a sabiendas, de que sería imposible implementarlo.

Resumiendo, creemos que hemos podido pecar de ser algo ambiciosos con querer trabajar con tecnologías muy punteras en el sector del desarrollo del Software, pese a no tener experiencia con la mayoría de ellas y eso nos está volviendo en forma de problemas, bugs y retraso en la planificación inicial.

**8-Listado de Acrónimos**

Abreviatura	Significado

**9-Referencias y Bibliografía**

- [1]
- [2]
- [3]
- [4]
- [5]
- [6]
- [7]
- [8]
- [9]

Anexo





