

Informe Inicial TFG

Aplicación de Gestión Financiera

Rafael Gómez Pérez
Mención en Ingeniería del Software



Índice de contenidos

1.	<u>Histórico de Revisiones</u>	2
2.	<u>Descripción del Problema</u>	3
3.	<u>Descripción de la Aplicación</u>	3
4.	<u>Objetivos del Proyecto</u>	4
5.	<u>Metodología empleada</u>	4
6.	<u>Planificación del problema</u>	5
7.	<u>Bibliografía y Referencias</u>	5
8.	<u>Listado de Acrónimos</u>	6

Histórico de revisiones

Fecha	Versión	Descripción	Autor
02/03/2019	1.0	Cambios en el formato del documento	Rafa
06/03/2019	1.1	Planificación inicial y bibliografía creadas, retoques al formato según feedback del tutor hechos.	Rafa
06/03/2019	1.2	Diagrama de Gantt añadido y reordenación de apartados	Rafa
07/03/2019	1.3	Nuevas referencias y listado de acrónimos añadidos	Rafa

Descripción del problema a resolver

El proyecto a realizar es una aplicación híbrida con la herramienta de desarrollo de aplicaciones móviles multiplataforma Ionic [1] cuyo propósito será gestionar las deudas de dinero que aparecen en todos los grupos de amigos junto con las finanzas domésticas utilizando una arquitectura basada en microservicios.

La idea consiste en permitir a los usuarios poder gestionar sus gastos, tanto domésticos, como con amigos dándoles una herramienta fácil de usar y muy visual que les permita identificar donde se están gastando su dinero.

Descripción de la aplicación

En la parte del proyecto correspondiente a finanzas domésticas, podremos llevar una cuenta de los gastos mensuales separados por categorías (costes fijos, ingresos, gastos variables etc) y subcategorías de estos (alquiler, agua, luz, gas, restaurantes, ingresos extraordinarios). La aplicación nos permitirá realizar un balance de los gastos acumulados durante este mes, incluso permitirá comparar los gastos del mismo sector en los meses seleccionados (o todo el año).

En el apartado de finanzas compartidas, podremos crear múltiples grupos donde mostrará en todo momento el balance general de las deudas de todos sus miembros, un historial de los últimos eventos, pagos y una recomendación del miembro del grupo que debería pagar durante el próximo evento.

Debido al ser una aplicación móvil consta de dos proyectos que se han de desarrollar (Backend y Frontend).

En cuanto al stack utilizado para el desarrollo del proyecto front es:

- Ionic Framework [1] (Typescript [2])
- HTML5
- CSS3
- SASS
- NPM [3]

En proyecto en backend constará de las siguientes tecnologías:

- Spring MVC, Data, Security, Cloud, Boot
- Netflix OSS (Hystrix [4], Ribbon [5], Heureka [6])
- DB noSQL en el Cloud de Digital Ocean
- Maven como gestor de dependencias
- Hibernate como Object-Relational Mapping (ORM)

Ambos proyectos utilizarán Docker [7] para realizar Continuous Integration (CI) [8] así como Test Driven Development (TDD) [9]. En la parte de Quality Assurance (QA) se realizará Unit test con JUnit 5 y Mockito. (Dependiendo del ritmo del proyecto se puede añadir más prácticas de test).

Objetivos del proyecto (ordenados de mayor a menor prioridad)

- **tfg-obj-01:** Poner en práctica los conocimientos de desarrollo del software adquiridos durante la mención.
- **tfg-obj-02:** Profundizar mis conocimientos en cada fase del desarrollo (Requisitos/Diseño/Implementación/Testing/Despliegue).
- **tfg-obj-03:** Aprender herramientas estandarizadas en el sector del desarrollo del software
- **tfg-obj-04:** Utilizar estándares de Clean Code y Design Patterns durante todo el ciclo de vida del proyecto.
- **tfg-obj-05:** Crear una aplicación que permita gestionar las finanzas de manera fácil y eficaz
 - **tfg-obj-05.1:** Poder gestionar de forma fácil mis finanzas domésticas a través de la aplicación.
 - **tfg-obj-05.2:** Poder gestionar las finanzas de diferentes grupos de amigos con la aplicación.
- **tfg-obj-06:** Realizar una arquitectura basada en microservicios y clustering en el cloud.
- **tfg-obj-07:** Crear una app multiplataforma (iOS y Android).

Metodología utilizada

Dado el número de personas que desarrollarán la aplicación, la mayoría de metodologías usadas a nivel empresarial o dadas durante la mención no servían. La mayoría de metodologías ágiles están pensadas para ser implementadas en el ámbito laboral y, por lo tanto, constan de una serie de roles dentro del equipo de desarrollo y/o actividades que han de realizarse de forma periódica.

Dado que en el equipo de desarrollo del proyecto solo cuenta con una persona, y que normalmente una persona no puede ejercer mas de un rol a la vez, metodologías como Scrum, Six Sigma o Lean no son aplicables a mi proyecto. De modo que he decidido utilizar Kanban[8] junto con feature-oriented programming[9] (FOP).

FOP es una metodología orientada al desarrollo incremental de un proyecto según las funcionalidades necesarias en cada momento. FOP usa unas métricas matemáticas para realizar los incrementos que, para mi proyecto, solo complicaría su puesta en práctica por lo que usaré la parte de selección de funcionalidades solamente.

Kanban es una metodología de desarrollo basada en un backlog de tareas que, a medida que las vamos seleccionando, van pasando por diferentes estados del dashboard (Backlog-New-Wip-Testing-Complete). A diferencia de Scrum, no hay un desarrollo en períodos de tiempo (sprints) ni roles dentro del equipo lo que lo hace más adaptable a cambios durante el proceso de desarrollo.

Kanban nos permite ponerle un límite máximo de tareas a cada estado para ayudar a detectar bottlenecks durante el desarrollo.

Planificación del proyecto

Debido a la naturaleza del proyecto, será separado en dos partes (Frontend y Backend). Puesto que se pretende generar un incremento en el software para ir generando demos, ambos proyectos han de ser desarrollados en paralelo según las necesidades de la funcionalidad que se incluirá en la siguiente demo.

Puesto que el proyecto se forma de dos grandes bloques de funcionalidades (finanzas domésticas y grupales), se ha decidido tratarlo como dos bloques completamente diferentes de la aplicación que serán desarrolladas secuencialmente.

Como el bloque de gestión de finanzas domésticas es el mas complejo, será desarrollado primero utilizando las metodologías mencionadas. Una vez haya sido completado, se procederá a implementar el bloque de finanzas grupales.

En el anexo, se encuentra el diagrama de Gantt acorde a la planificación mencionada.

Referencias y bibliografía

- [1] <https://ionicframework.com/>, Febrero 2019
- [2] <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>, Marzo 2019
- [3] <https://www.npmjs.com/>, Marzo 2019
- [4] <https://www.baeldung.com/introduction-to-hystrix>, Marzo 2019
- [5] <https://www.paradigmadigital.com/dev/ribbon-equilibrador-cargas-peticiones-eureka/>, Marzo 2019
- [6] <https://www.baeldung.com/spring-cloud-netflix-eureka>, Marzo 2019
- [7] <https://docs.docker.com/get-started/>, Marzo 2019
- [8] https://en.wikipedia.org/wiki/Continuous_integration, Marzo 2019
- [9] https://en.wikipedia.org/wiki/Test-driven_development, Marzo 2019
- [10] <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban/> Marzo 2019
- [11] https://en.wikipedia.org/wiki/Feature-oriented_programming, Marzo 2019

Listado de Acrónimos

Abreviatura	Significado
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SASS	Syntactically Awesome Stylesheets
NPM	Node Package Manager
ORM	Object-Relational Mapping
CI	Continuous Integration
TDD	Test Driven Development
QA	Quality Assurance
FOP	Feature Oriented Programming

Anexo

