

IFESOF

# MEMORIA

2017-2018

Rubén García, Iván Fernández, Alberto Pastor, Gerardo  
Parra, Pablo López y Marco Adinolfi

# Contenidos

1	Introducción.....	2
1.1	Contenido de memoria.....	2
1.2	Objetivos .....	2
2	Planificación .....	2
2.1	Planificación inicial.....	2
2.2	Planificación final.....	2
3	Conceptos previos.....	3
4	Estructura y desarrollo.....	3
5	Pruebas.....	4
6	Conclusiones.....	4

# 1 INTRODUCCIÓN

---

## 1.1 CONTENIDO DE MEMORIA

Esta memoria contiene las etapas y el tratamiento que ha seguido el desarrollo del proyecto software con el nombre IFESOFT. El proyecto consiste en un programa informático para gestionar ferias, salones y congresos en sus instalaciones, las cuales divididas en pabellones. En sus eventos se dan cita empresas españolas y extranjeras para generar relaciones comerciales.

Este documento contiene los objetivos, la planificación, estructura y el modo de desarrollo del proyecto. Se tratan los cambios que se han ido gestionando durante la creación del producto final.

## 1.2 OBJETIVOS

*El objetivo que presenta la realización de este proyecto es que las empresas se beneficien de nuestro software pudiendo gestionar todas las funciones especificadas en los requisitos. Nuestro usuario final es el gestor o encargado de la empresa organizadora de los eventos.*

# 2 PLANIFICACIÓN

---

## 2.1 PLANIFICACIÓN INICIAL

Durante una primera estimación del proyecto se planifico distribuir el trabajo entre siete personas siguiendo una estructura descentralizada democrática dividiéndonos ocasionalmente en dos subgrupos de trabajo para ocuparse de módulos diferentes del proyecto.

Se eligió esta estructura por ser un grupo de trabajo pequeño, creyendo conveniente tomar las decisiones juntos y no hacer reuniones específicas solo para los jefes de los dos subgrupos.

En cuanto a planificación de trabajo se concretó que cada integrante trabajaría una hora diaria, descansando los sábados. De tal forma que se finalizaría un módulo por semana, incluyendo revisión y validación con el profesor.

## 2.2 PLANIFICACIÓN FINAL

Durante el desarrollo del software se hizo real uno de los riesgos analizados en el plan de proyecto.

Algunos integrantes del equipo no cumplieron el trabajo asignado durante varias etapas. Esto conllevó el incumplimiento de la planificación obligando a aumentar las horas de trabajo del resto del equipo, llegando a retrasar algunas entregas.

Además, la estructura del equipo se reestructuró en equipos especializados por capas y con responsabilidades internas en cada sub-equipo. Con el fin de aumentar el alcance de trabajo de cada integrante, de ahí se consiguió terminar el trabajo a tiempo.

Como consecuencia del cambio en la forma de trabajo se tomó la decisión de realizar una implementación directa con un diseño poco detallado dando libertad a los programadores y realizando un diseño basado en la implementación final del código.

### 3 CONCEPTOS PREVIOS

---

La capacidad técnica del equipo se vio limitada por la falta de conocimientos, ya que en algunas partes del proyecto eran necesarias nuevas competencias que fueron obtenidas durante el transcurso del desarrollo del proyecto.

El equipo se ha tenido que formar en las siguientes tecnologías: JDBC, MariaBD y su API, Java Swing, IntelliJ IDE, herramientas de SCV y el uso de herramientas CASE, en particular IBM RSA. Sin olvidar los conocimientos específicos que proporciona la asignatura.

### 4 ESTRUCTURA Y DESARROLLO

---

El proyecto sigue una arquitectura multicapa y está estructurado en paquetes correspondientes a los módulos representados en el modelo del dominio. En particular la capa de presentación se encuentra organizada por funciones u operaciones para mejorar la comprensión por parte del usuario final.

Para el diseño arquitectónico utiliza los siguientes patrones: **Singleton**, **factorías abstractas y controller** y para representar las entidades del modelo del dominio se han utilizado los patrones **transfer**, **DAO** y los **servicios de aplicación**.

Para la capa de recursos se ha utilizado **MariaBD y su SGBD**.

Como herramientas de control de versiones hemos utilizado el **Tortoise SVN** y un repositorio de **Git**. Para código se ha utilizado el repositorio de Git y para los diseños utilizando la herramienta CASE se ha usado SVN.

## 5 PRUEBAS

---

Hemos realizado dos tipos de pruebas: caja negra y caja blanca. Las pruebas de caja negra se han validado desde la capa de presentación y las pruebas de caja blanca se han verificado a través del **framework JUnit**.

Se han realizado también pruebas unitarias, es decir, de cada método para comprobar así su correcto funcionamiento.

## 6 CONCLUSIONES

---

Como conclusión, el equipo ha llegado a las siguientes reflexiones:

- A pesar de haber realizado análisis de riesgos estrictos y desarrollar planes de contingencia, el equipo no ha sido capaz ponerlos en práctica correctamente. Esto ha causado retrasos en la planificación y problemas internos en el equipo.
- La falta de conocimientos de la herramienta CASE y el sistema de control de versiones SVN nos llevó a múltiples retrasos inesperados y pérdida de parte del trabajo realizado previamente, forzándonos a aumentar las horas de dedicación.
- La metodología de trabajo del equipo ha derivado a lo largo del desarrollo del proyecto software en la especialización de los miembros del equipo por capas, mejorando así la eficiencia en el flujo de trabajo, pero generando la necesidad de un aumento de la carga de trabajo para adquirir las competencias completas de la asignatura.
- La implementación de un diseño inicial poco detallado, así como la implementación directa de dichos diseños a código ha forzado a llevar a cabo una gestión del cambio más fluida, sin embargo, en casos puntuales ha conducido a enfrentar grandes aumentos en la carga de trabajo.

Como conclusión los integrantes del equipo han obtenido experiencia en todos los ámbitos del desarrollo software, en especial, en la gestión del trabajo en equipo.