

GREENGO

# MEMORIA

2018-2019

Pablo Agudo, Alejandro Cabezas, Iván Fernández,  
Manuel Monforte, Gerardo Parra y Alberto Pastor

# 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 GREENGO. La aplicación por desarrollar recibirá el nombre de "GREENGO", la cual llevará a cabo la gestión de distintos vehículos operativos en ciudades concretas, permitiendo el alquiler de estos por clientes registrados previamente, así como la gestión de diferentes sedes, servicios contratados y empleados.

Dentro de la aplicación, gestionaremos seis módulos principales que son Vehículos, Ciudades, Clientes, Sedes, Servicios y Empleados. Dentro del modulo de vehiculo diferenciaremos dos tipos, los coches y las bicicletas.

Este documento contiene los objetivos, la planificación, estructura y el modo de desarrollo del proyecto. Se tratan los cambios que se han gestionado 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 seis personas siguiendo una estructura descentralizada democrática dividiéndonos ocasionalmente en tres subgrupos de trabajo para ocuparse de capas diferentes del proyecto.

La decisión de tener tres equipos especializados por capas y con responsabilidades internas en cada subequipo se realizo con el fin de aumentar el alcance de trabajo de cada integrante, lo cual nos facilitó terminar el trabajo a tiempo.

Además se uso como referencia la experiencia de ambos equipos en la realizacion de los proyectos de la asignatura con nombre Ingeniería del Software, decidiendo así la separacion por capas de los equipos.

También elegimos esta estructura por ser un grupo de trabajo pequeño, creyendo conveniente tomar las decisiones juntos y no llevar a cabo reuniones en las cuales solo participaran los jefes de los tres subgrupos.

En cuanto a planificación de trabajo concretamos que cada integrante trabajaría una hora diaria, descansando los sábados. Se realizarían requisitos,

diseño, TDD (desarrollo guiado por *testing*) y codificación en este orden para la correcta realización del proyecto.

## 2.2 PLANIFICACIÓN FINAL

Durante el desarrollo del software no se hicieron reales los riesgos que indicamos en los requisitos del sistema.

En la realización del proyecto se a seguido toda planificación esperada menos dos puntos que el equipo tuvo que subsanar:

- Falta de tratamiento en el sistema de las fechas asociadas al modulo Rental.
- Falta de revisión de los tests (TDD) para su uso correcto en la etapa de codificación.

Todos estos inconvenientes se subsanaron con un aumento de las horas de trabajo en las últimas etapas del desarrollo del sistema.

## 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 especialmente en los conocimientos nuevos necesarios de la asignatura de Modelado de Software. Aunque en la asignatura Ingeniería del Software se consiguieron los conocimientos básicos de las tecnologías necesarias para el proyecto, muchas de ellas se han tenido que revisar.

Estas tecnologías son: JDBC, MariaBD y su API, Java Swing, IntelliJ IDE, herramientas de SCV y el uso de herramientas CASE, en particular IBM RSA.

## 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 hemos utilizado los siguientes patrones: **Singleton**, **factorías abstractas**, **controller**, **commands**, **dispatcher view** y para representar las entidades del modelo del dominio se han utilizado los patrones **transfer**, **TOAS**, **DAO** y los **servicios de aplicación**.

Se ha seguido el patrón **Almacen de dominio** para la realización del sistema.

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:

- Los integrantes del equipo se han dado cuenta de la importancia de la realización de TDD. Con esta metodología de trabajo se localizaron antes los errores y se trabajó de manera más eficiente.
- Aunque la importancia de los requisitos y del diseño ha estado presente en todo el desarrollo, hemos comprendido cómo agiliza tener estos dos aspectos del proyecto bien entendidos y definidos.
- Como en proyectos con tiempos de entrega cortos y equipos pequeños es necesario que todos los miembros trabajen lo suficiente y aproximadamente a la misma velocidad. Este año se ha reflejado esta buena conducta en el equipo.

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.