Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Analysis Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2023 – 2024

| **Fecha** | **Versión** |
| --- | --- |
| 16/02/2024 | v1.0.0 |

| **Grupo: C1.011** | | |
| --- | --- | --- |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Abouri, Mohamed - Y7156458E | Developer | Persona encargada de desarrollar el código. |
| Cabello Ranea, Rafael -32094496C | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa. |
| Calderón Rodríguez, Francisco Javier - 78233219F | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| Delgado Pallares, David - 29519510E | Tester | Persona encargada de realizar pruebas sobre el código. |
| Ramírez Lara, Iván- 77852382G | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

| **Fecha** | **Versión** | **Descripción** |
| --- | --- | --- |
| 06/03/2024 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

**Índice de contenido**

[1.](#_heading=h.gjdgxs) Resumen ejecutivo 2

2[.](#_heading=h.gjdgxs) Introducción 2

3[.](#_heading=h.30j0zll) Evaluación 2

4[. Conclusiones](#_heading=h.1fob9te) 6

**1. Resumen ejecutivo**

Este documento tiene el objetivo de hacer una retrospectiva de todas las tareas que hemos establecido que vamos a realizar en el planning report. Esta retrospectiva nos servirá para descubrir qué tareas han ido bien, mal o regularmente. De esta forma podremos llevar a cabo un plan de mejora con el objetivo de evitar estos errores en entregas posteriores.

**2. Introducción**

Habiendo realizado ya todas las tareas que hemos establecido del sprint planning, es hora de ver cómo ha ido el desarrollo de estas y qué plan de mejora podemos establecer para el siguiente deliverable, evaluando los pros y los contras.

**3. Evaluación**

En esta entrega, hemos corregido el problema de dejar algunas tareas para el final del entregable como requisitos opcionales de documentación. Gracias al plan de mejora que implementamos en el sprint anterior, hemos podido llevar la documentación al día, desarrollando al mismo tiempo que el código.

En segundo lugar, en este entregable no se han realizado bien las estimaciones de ciertas tareas, ya que no se ha establecido un tiempo de contingencia en caso de error en algunas de las tareas. En este entregable se ha tenido que emplear tiempo extra en estas tareas:

* Corrección de la clase Client
* Corrección de la clase Objective
* Corrección del Diagrama uml

Para que esto no ocurra en los siguientes sprints, hemos desarrollado un plan de mejora en el que hemos establecido que se tiene que estar seguro de que la clase está perfectamente implementada y que no tiene ningún error que se tiene que corregir antes de hacer el merge con la rama master. Para ello hemos decidido crear una rama Test/feature desde Master y hacer merge con ella en vez de Master. Así podemos asegurarnos de que todo va bien antes de mergear con Master.

Pros: Cuando se acepte la pull request se estará seguro de que la implementación es correcta.

Contras: Tenemos que tomar un poco más de tiempo en implementar y testear la nueva funcionalidad para que no de error y tardaremos más tiempo en hacer merge con las nuevas características.

**4.** [**Conclusiones**](#_heading=h.1fob9te)

En este entregable hemos analizado el plan de mejora que habíamos implementado en la entrega anterior y hemos concluido que ha sido de gran ayuda para realizar las diferentes tareas del entregable. Con respecto a los problemas que hemos encontrado en este entregable, podemos decir que no han sido graves, ya que solo nos han causado un poco de retraso. Para intentar reducir esto, en el siguiente entregable usaremos el plan de mejora y lo analizaremos para ver cómo ha funcionado.