

Cuidando a Pancho



UNIVERSIDAD
POLITÉCNICA
DE MADRID

Caso de Estudio: Cuidando a Pancho

**Fundamentos de Ingeniería del
Software 2023-24**

UPM - ETSISI

Valoración Crítica

1.INTRODUCCIÓN

-El trabajo grupal de la asignatura de Fundamentos de Ingeniería del Software consistía en elaborar una aplicación de gestión de cuidado de mascotas, llamada “Cuidando a Pancho”. Para ello, se proporcionó un documento donde se explicaba el funcionamiento de dicha aplicación y los distintos requisitos que se exigían en esta. Y se ordenaba realizar dicha aplicación siguiendo los conceptos y fases desarrolladas en la asignatura.

2.FASES DE DESARROLLO

-Para la realización del proyecto, se han seguido una serie de fases para facilitar su implementación desde el enunciado de este hasta su implementación en código, que son las siguientes:

- **ESPECIFICACIÓN DE REQUISITOS:** En esta fase, el objetivo es entender y comprender el funcionamiento completo del proyecto y plasmarlo en una serie de requisitos descritos en RedMine siguiendo el estándar IEEE 830. Esta fase nos ha permitido entender de primeras el proyecto para así poder pensar en las siguientes fases con una idea completa y sólida sobre este.
- **FASE DE ANÁLISIS:** En esta fase, el objetivo es desarrollar los diagramas de casos de uso y con ellos el diagrama de clases de análisis partiendo de los requisitos especificados en la fase anterior. Con los requisitos, como he mencionado se desarrollan los diagramas de casos de uso que nos dan una idea de como interaccionan los usuarios con las distintas funcionalidades de la aplicación, y con ellos poder desarrollar los diagramas de clases de diseño, que son muy útiles para poder concebir las primeras ideas sobre las clases que la aplicación necesitará en su etapa de implementación.
- **FASE DE DISEÑO:** En esta fase, a través de los diagramas de clases de análisis se obtienen los diagramas de clases de diseño, que ya asientan las bases para poder empezar a codificar el programa. Esta fase no permite poder refinar las ideas descritas en el diagrama de clases de análisis y con ello poder empezar a pensar en métodos que son necesarios codificar para que cumplan las funcionalidades solicitadas para el proyecto.
- **FASE DE IMPLEMENTACIÓN:** En esta fase, ya con el diagrama de clases de diseño hecho, se empieza a desarrollar el código de la aplicación, permitiendo poder hacer una generación automática de código desde los diagramas de clases de diseño que nos facilita el trabajo y nos ahorra mucho tiempo en su implementación. Esta fase se puede considerar la más compleja de todas, ya que pueden ir surgiendo distintos problemas en la implementación del código, y se necesitará ajustar el código de distintas maneras para que satisfaga las necesidades que requiere el proyecto.
- **FASE DE PRUEBAS:** Una vez acabado el código, se comienza con esta fase, que sería la última en este proyecto. En donde es necesario comprobar que el código se adapta correctamente a las necesidades descritas en el enunciado del proyecto, para ello se hacen unas pruebas unitarias, para comprobar que los métodos referidos a la creación de usuarios funcionan correctamente y no dan fallos, mediante pruebas de caja negra con Junit. Tras esto se realizan las pruebas de aceptación de la creación de nuevos usuarios, en donde se necesita relacionar la ejecución del código con los requisitos funcionales descritos en RedMine.

3.CONCLUSIÓN

-Las fases de análisis, diseño, implementación y pruebas han sido cruciales para la finalización exitosa del proyecto.

- **ESPECIFICACIÓN DE REQUISITOS:** La fase de especificación de requisitos fue fundamental para el proyecto. Utilizando RedMine y el estándar IEEE 830, documentamos todas las necesidades del proyecto. Esta claridad inicial aseguró que todas las fases posteriores se alinearan con las expectativas del cliente, facilitando un desarrollo coherente y enfocado desde el principio.
- **FASE DE ANÁLISIS:** La fase de análisis nos permitió entender cómo los usuarios interactuarían con el sistema a través de diagramas de casos de uso y clases de análisis. Esto facilitó una base sólida para el diseño, asegurando que todas las funcionalidades estuvieran claramente definidas desde el principio.
- **FASE DE DISEÑO:** En la fase de diseño, transformamos los diagramas de análisis en diagramas de clases de diseño detallados. Esto estructuró el código de manera organizada, identificando métodos y relaciones entre clases, lo que preparó el terreno para una implementación eficiente.
- **FASE DE IMPLEMENTACIÓN:** Durante la fase de implementación, desarrollamos el código basándonos en los diagramas de diseño. Esta etapa fue la más compleja, pero el trabajo detallado en las fases anteriores facilitó la resolución de problemas y permitió un desarrollo más fluido.
- **FASE DE PRUEBAS:** Finalmente, la fase de pruebas verificó que el software cumpliera con los requisitos especificados. Las pruebas unitarias y de aceptación aseguraron que los métodos funcionaran correctamente y que el sistema en su totalidad se alineara con los requisitos funcionales.

En resumen, cada fase contribuyó significativamente a la finalización del proyecto, garantizando una implementación del código organizada, eficiente y de alta calidad.