Analysis Report

C2.017

https://github.com/vicgrabru/Acme-SF-D04

Carlos García Ortiz
cargarort3@alum.us.es

Tabla de contenidos

Resumen ejecutivo

El análisis de requisitos es un paso fundamental en el proceso de desarrollo de un sistema, ya que permite identificar y documentar las necesidades y expectativas de los usuarios y las partes interesadas. Un análisis de requisitos completo y preciso ayuda a evitar deuda técnica de cara al futuro del desarrollo.

Es importante involucrar a los usuarios y partes interesadas para comprender sus necesidades y expectativas, y documentar estos requisitos de manera clara y concisa. De esta manera, el equipo de desarrollo puede diseñar e implementar un sistema que cumpla con estos requisitos y expectativas.

En general, un análisis de requisitos efectivo es esencial para el éxito de un proyecto de desarrollo de sistemas.

Tabla de revisiones

Revisión	Fecha	Descripción
1.0	27-06-2024	Primera versión del documento
2.0	02-07-2024	Versión final del documento

Introducción

En este documento ofrece una visión detallada de los problemas identificados durante la interpretación y el enfrentamiento de los requisitos del estudiante 2, así como las decisiones tomadas al respecto y las conclusiones obtenidas.

Por cada caso, se presentará el requisito en cuestión y el análisis realizado, del que se aportarán las conclusiones detalladas, decisiones tomadas y enlace a la validación de los profesores en el foro, si procede.

Contenido

Relación entre borradores de contracts y progress logs

Como se comentó en el análisis del entregable anterior, contracts y progress logs son entidades padre-hijo cuya única restricción es que no se pueden crear progress logs si el contract no está publicado.

Debido a esta restricción algunos tests son más simples que los de otras entidades padre e hijo, por ejemplo el delete.safe sólo requiere borrar un contract ya que sólo existe el caso de que un contract en borrador no tenga hijos. Igualmente se han hecho los tests obteniendo el máximo coverage posible teniendo esto en cuenta.

Money Exchange invariante

Durante los tests de create.safe y update.safe de la entidad contract se tuvo que probar en el campo presupuesto un caso positivo usando una divisa aceptada por el sistema que fuese diferente a la del sistema. Usamos en este caso el dato “USD 0.00” con la intención de que en cualquier momento que se se repitan los tests el resultado no varíe junto al ratio obtenido por api.

Igualmente esto es sólo una segunda medida de protección ya que, como se comentó en el análisis grupal, hemos mockeado el método que llama a la api para que durante testing siempre devuelva ratio 1.

Conclusión

A la hora de testear hay que tener en cuenta muchas cosas antes de lanzarse a grabar, pues sino se planean las cosas de antemano los tests se volverán una completa pesadilla y habrá que volver a hacer muchos de ellos una y otra vez por cada piedra que no se vio desde lejos.

Al final 2 requisitos terminan siendo más complicados que todos los anteriores entregables si no analizas bien la situación.

Bibliografía

No aplica.