



Analysis report

C2.03.14

<https://github.com/c1-03-14/c1.03.14>

Victor Graván Bru

vicgrabru@alum.us.es

Alberto Antonio Tokos Mata

albtokmat@alum.us.es

Gonzalo Santiago Martín

gonsanmar2@alum.us.es

Fernando Barroso Barroso

ferbarbar1@alum.us.es

Alvaro Sánchez Flores

alvsanf1o1@alum.us.es

Tabla de contenidos

Tabla de contenidos	1
Resumen ejecutivo	2
Tabla de revisiones	2
Introducción	3
Contenido	4
Conclusión	8
Bibliografía	8

Resumen ejecutivo

El análisis de requisitos es un paso fundamental en el proceso de desarrollo de un sistema, ya que es el momento en que se identifican y documentan las necesidades y expectativas de los usuarios y las partes interesadas.

Un análisis de requisitos completo y preciso puede ayudar a evitar problemas y cambios costosos más adelante en el proceso de desarrollo del sistema. Es importante involucrar a los usuarios y partes interesadas en el proceso de análisis de requisitos para comprender sus necesidades y expectativas, y documentar estos requisitos de manera clara y concisa.

De esta manera, el equipo de desarrollo puede trabajar para 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	03-07-2023	Primera versión del documento
1.1	07-07-2023	Arreglos finales

Introducción

La fase de análisis de requisitos es un paso fundamental en el proceso de desarrollo de un sistema, ya que es el momento en que se identifican y documentan las necesidades y expectativas de los usuarios y las partes interesadas. El análisis de requisitos se enfoca en comprender los objetivos del sistema, los procesos que se deben llevar a cabo, los datos que se deben capturar y los resultados que se esperan del sistema.

Para llevar a cabo un análisis de requisitos efectivo, es necesario involucrar a los usuarios y partes interesadas para entender sus necesidades y expectativas, y luego documentar estos requisitos de manera clara y concisa. De esta manera, el equipo de desarrollo puede trabajar para diseñar e implementar un sistema que cumpla con estos requisitos y expectativas.

Un análisis de requisitos completo y preciso puede ayudar a evitar problemas y cambios costosos más adelante en el proceso de desarrollo del sistema. Por lo tanto, es importante dedicar suficiente tiempo y recursos para el análisis de requisitos para garantizar que el sistema cumpla con las necesidades y expectativas de los usuarios y partes interesadas, y tenga éxito en su implementación.

Contenido

Requisitos para casos de prueba para los requisitos #14y # 15.

Descripción:

El requisito establece la necesidad de crear una suite de pruebas para verificar y validar los requisitos relacionados con las operaciones que los profesores pueden realizar en los cursos y/o en las clases. A continuación, se analiza el enfoque general para diseñar una suite de pruebas efectiva:

Requisitos:

Identificar los casos de prueba: En primer lugar, se deben identificar los diferentes casos de prueba que cubrirán los escenarios clave relacionados con los requisitos. Por ejemplo, se pueden considerar casos de prueba para listar los cursos o las clases, mostrar los detalles de un curso o una clase, crear y publicar una clase, y actualizar o eliminar una clase no publicada.

Diseñar casos de prueba: Para cada caso de prueba identificado, se debe diseñar un conjunto de pasos detallados y específicos que permitan verificar la funcionalidad requerida. Esto incluye definir los datos de prueba necesarios, las acciones a realizar y los resultados esperados. Por ejemplo, para el caso de prueba "Mostrar detalles de una clase", los pasos pueden incluir iniciar sesión como profesor, seleccionar una clase específica y verificar que se muestren correctamente los detalles esperados.

Preparar datos de prueba: Para ejecutar los casos de prueba, es posible que sea necesario preparar datos de prueba, como crear cursos y clases ficticios en el sistema, asignar un profesor a los cursos, o establecer el estado de publicación de las clases según los casos de prueba específicos.

Ejecutar los casos de prueba: Una vez que se hayan diseñado los casos de prueba y preparado los datos necesarios, se deben ejecutar los casos de prueba y registrar los resultados. Durante la ejecución, es importante asegurarse de seguir los pasos definidos y comparar los resultados obtenidos con los resultados esperados.

Analizar los resultados y corregir errores: Después de ejecutar los casos de prueba, se deben analizar los resultados y compararlos con los resultados esperados. Si se encuentran discrepancias o errores, se deben documentar y reportar para que puedan ser corregidos por los desarrolladores.

Repetir el proceso: La suite de pruebas debe ser iterativa, lo que significa que se puede modificar y mejorar a medida que se descubran nuevos casos de prueba o se realicen cambios en los requisitos. Es importante seguir repitiendo el proceso de diseño, ejecución y análisis de pruebas para garantizar la calidad y la conformidad con los requisitos.

Conclusión:

Por lo tanto, el requisito establece la necesidad de producir una suite de pruebas para verificar los requisitos relacionados con las operaciones de los profesores en los cursos y/o en las clases. Siguiendo el enfoque mencionado anteriormente, se puede diseñar y ejecutar una suite de pruebas efectiva que permita validar la funcionalidad requerida y garantizar que los requisitos se cumplan de manera adecuada.

Análisis de requisito:

Este requisito estaba bien definido y no representa ambigüedad o incompletitud, no es necesario analizar este requisito.

Requisitos de datos para los casos de prueba.**Descripción:**

El requisito establece la necesidad de generar datos de prueba variados para enriquecer la suite de pruebas. Estos datos de prueba se utilizan para probar diferentes escenarios y casos de prueba en la suite de pruebas. A continuación, se analiza cómo generar datos de prueba variados para la suite de pruebas:

Requisitos:

Identificar escenarios de prueba: Es importante identificar los diferentes escenarios de prueba que se cubrirán en la suite de pruebas. Esto implica comprender los requisitos, las funcionalidades y los casos de uso del sistema. Por ejemplo, si se están probando las operaciones de los profesores en los cursos, se pueden identificar escenarios como la creación de un nuevo curso, la actualización de un curso existente y la eliminación de un curso.

Generar datos de prueba relevantes: Para cada escenario de prueba, se deben generar datos de prueba que sean relevantes y representativos del escenario. Esto puede incluir la creación de cursos ficticios con diferentes características, como títulos, resúmenes, duraciones, niveles, etc. Además, se pueden generar datos de prueba para simular diferentes estados de publicación, como cursos publicados y no publicados, así como también clases asociadas a esos cursos.

Considerar casos límite y casos de error: Es importante generar datos de prueba que cubran los casos límite y los casos de error. Esto implica probar con valores extremos o límites de los campos de entrada, como títulos muy largos o muy cortos, resúmenes vacíos, duraciones negativas, entre otros. También se deben generar datos de prueba que simulen situaciones de error, como cursos o clases inexistentes o inaccesibles.

Utilizar datos de muestra representativos: Además de generar datos ficticios, también se pueden utilizar datos de muestra representativos para realizar pruebas más realistas. Esto puede incluir datos reales de cursos, clases y profesores que hayan sido anonimizados y utilizados con fines de prueba.

Documentar y organizar los datos de prueba: Es importante documentar y organizar los datos de prueba de manera clara y estructurada. Esto facilitará su uso y referencia durante la

ejecución de la suite de pruebas. Se pueden utilizar hojas de cálculo, archivos de texto u otras herramientas para registrar y mantener los datos de prueba.

Conclusión:

De este modo, el requisito establece la necesidad de generar datos de prueba variados para enriquecer la suite de pruebas. Al identificar escenarios de prueba, generar datos relevantes, considerar casos límite y casos de error, y utilizar datos de muestra representativos, se puede producir una variedad de datos de prueba que cubran diferentes situaciones y escenarios. Estos datos de prueba ayudarán a garantizar una cobertura exhaustiva de los casos de prueba y una validación más efectiva del sistema.

Análisis de requisito:

Este requisito estaba bien definido y no representa ambigüedad o incompletitud, no es necesario analizar este requisito.

Conclusión

Este entregable el único inconveniente que tenía era que antes de poder realizar estas pruebas debíamos tener terminado bien todo lo expuesto en los 3 anteriores sprints. Por lo demás no había ningún tipo de ambigüedad ni duda, ya que, es realizar un conjunto de tests para probar que funcione correctamente toda la aplicación.

Bibliografía

Intencionalmente en blanco.