

INFORME DE ANÁLISIS

GRUPO: C1.02.11

Juan Luis Ruano Muriedas (juaruamur@alum.us.es)

Índice

1.	Resumen ejecutivo	1
2.	Historial de versiones	2
3.	Introducción	2
4.	Contenidos	3
	4.1. Listado de registros de análisis	3-4
5.	Conclusiones	4
6.	Bibliografía	4

1. Resumen ejecutivo

En este reporte se desarrollará toda la información referente al análisis de los requisitos más complejos de cada uno de los entregables.

Para ello, contaremos con una copia literal del requisito al que se refiere el registro y daremos las conclusiones detalladas del análisis y decisiones tomadas para subsanar dicho requisito, si se diera el caso, aportamos un enlace a la validación realizada por un profesor.

2. Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
15/02/2023	1.0	Creación de los documentos para el entregable 1	1

3. Introducción

En primer lugar, decir que en este documento se analizarán y explicarán cada uno de los requisitos abarcados por los entregables.

Los pasos a seguir para afrontar este análisis será leer detenidamente todos los requisitos del nuevo delivery, hecho esto, tendremos que distinguir entre dos tipos de requisitos:

Por una parte, los requisitos que por su simplicidad o bien por ser de los primeros en el comienzo de este proyecto no necesitaran un análisis previo.

Por otra parte, los requisitos de código que por su dificultad necesitaran de un análisis para su comprensión y valoración de cómo empezar a desarrollarlos. Es evidente que cuando vayamos avanzando en el proyecto los requisitos de las entregas empezarán a complicarse cada vez más.

El análisis exhaustivo de estos requisitos será muy importante ya que este nos dará una idea previa de lo que tendrá que hacer el desarrollador que le toque realizar esa tarea, viéndose beneficiado el tiempo como el coste de la misma.

Además, nos permitirá realizar una repartición más equilibrada de los requisitos ya que al conocer la dificultad, balanceamos la carga de trabajo para que unos compañeros no realicen más trabajos que otros.

En definitiva, creo que este documento nos ayudará a todos los integrantes del grupo tanto a estar en contexto de los requisitos más difíciles del sprint, como a conocer las conclusiones detalladas del análisis y las decisiones tomadas para subsanar el requisito.

4. Contenidos

En este punto encontraremos dos partes diferenciadas del requisito que será analizado:

- Una copia literal del requisito al que se refiere el registro.
- Conclusiones detalladas del análisis y decisiones tomadas para subsanar el requisito.

4.1. Listados registro de análisis

FR01 Operaciones de usuarios anónimos en cuentas de usuario:

- Regístrate en el sistema y conviértete en asistente.

Análisis y toma de decisiones

Este requisito consiste en que cualquier usuario registrado pueda convertirse en asistente y acceder a todas las opciones de tutoriales. Para ello hubo que hacer la vista de registro de asistente para introducir todos los datos del asistente en un formulario y el mismo formulario servirá para actualizarlo más adelante. Para que todo este proceso sea exitoso habrá que crear los componentes del registro, las clases Controller, Repository, Create Service y Update Service del paquete

authenticated.assistant. Controller maneja si se trata de una petición de crear o de modificar, mientras que repository se encarga de acceder a la base de datos y los service de que manejen las funciones de crear el asistente y modificar los datos del asistente. Todo esto está basado en la misma estructura que en el proyecto de ejemplo de acme jobs y me ayudó de las transparencias para que me ayudaran con los métodos de los service. La estructura de las clases Service son similares entre distintos apartados así que voy a elaborar en la clase Service aquí para que sirva para más adelante. Por lo general los métodos service se dividen en check, que comprueba el acceso a la página; authorise, que comprueba que el usuario tenga acceso al servicio en cuestión; load, que carga datos de la entidad de la base de datos a través de una llamada a la clase repository; bind, que crea la entidad para que acceda en el servicio; perform, que guarda datos en la base de datos llamando al repository y unbind, que se encarga de preparar los datos que se van a enviar al jsp de la vista. Para esta función se hizo todo esto junto a los formularios de registro como asistente en las vistas. El usuario solo pulsa en “hacerse asistente”, el controlador lo lleva al servicio del formulario y este se encarga de guardar los datos una vez introducidos. Para esta tarea me ayudó ver los ejemplos del acme jobs ya que también tiene asignación de roles y algunas correcciones del profesor.

FR02 Operaciones de asistentes en cuentas de usuario:

- Actualizar los perfiles.

Análisis y toma de decisiones

Este requisito es similar al anterior, simplemente se trataba de un servicio diferente, en vez de Create sería Update y basándose en la misma estructura y restricciones que para el create hice este apartado.

FR03 Operaciones de cualquier usuario autenticado en tutorías:

- Lista de tutorías asociadas a los cursos que se pueden listar.
- Enseña los detalles de la lista de tutorías que pueden cursar, incluyendo sus asistentes.

Análisis y toma de decisiones

Este requisito ya era algo diferente, trataba de sacar todos los datos de las tutorías y enseñarlos en una lista. Aquí se trata con el servicio List el cual enseña un conjunto de datos como lista. El servicio lista en sí era más sencillo que formulario pero requiere retocar más a fondo las vistas por el hecho de seleccionar qué botones aparecen, las columnas y traer la id al servicio show para enseñar los detalles al pulsar. Aquí me ayudaron las diapositivas de la asignatura que lo explican claramente.

FR04 Operaciones de asistentes en tutorías:

- Lista de tutorías que han creado.
- Enseña los detalles de sus tutorías.
- Crea, modifica o elimina las tutorías siempre que no hayan sido publicadas

Análisis y toma de decisiones

Aquí requería añadir un parámetro adicional “publicado” antes de empezar y por tanto retocar algo la base de datos. El servicio del formulario de crear y modificar lo hice de forma similar a un asistente ya que es la misma estructura. Cree un service delete para eliminar las tutorías y ajuste los botones dependiendo de si estaba publicado o no para que no se pudiera modificar ni borrar si estaba publicado. Para este solo tuve que fijarme en el acme jobs y en los service realizados anteriormente.

FR05 Operaciones de asistentes en sesiones de tutoría:

- Lista de sesiones por tutoría.
- Enseña los detalles de sus sesiones.
- Crea una nueva sesión en una tutoría, siempre que haya sido publicada.
- Modifica o elimina las sesiones en sus tutorías, siempre que no hayan sido publicadas.

Análisis y toma de decisiones

En este requisito se empezaba a trabajar con las sesiones que están relacionadas en grupos a cada tutoría. La mayor parte es parecida al requisito anterior pero requiere mucha más atención para trabajar en los Service ya que requieren arrastrar la id de la clase tutoría a la que pertenece, además hay que restringir según si está publicada o no la tutoría. Las sesiones tienen fuertes restricciones de tiempo que no conseguí acertar debido a que teníamos que trabajar con la versión 1.8 de java y la clase Date en vez de LocalDate y Date ya tenía varias clases deprecadas así como la 1.8 no tenía las mayores comodidades precisamente. Pero tras ayudarme el profesor de laboratorio, de las transparencias de clase y fijarme algo del proyecto Acme Jobs finalice la tarea.

FR06 Operaciones por asistentes en las estadísticas de asistente:

- Enseña las estadísticas de los asistentes.

Análisis y toma de decisiones

Este requisito fue el mayor desafío del entregable ya que requería hacer muchas llamadas a la base de datos de MariaDB con la que casi no había trabajado y además tenía que trabajar los periodos de tiempo con dos Timestamps lo cual no se hizo fácil. El service no se hizo tan complicado como largo pero para las llamadas en la clase Repository me ayudaron algo las transparencias y sobre todo mis compañeros de clase que tenían también query bastante complejas y me dieron direcciones al respecto.

5. Conclusión

Conclusivamente este documento nos va servir de apoyo para contar cómo hemos realizado los requisitos más complejos de los sprint, y detallar qué decisiones importantes hemos tomado y como se han solventado los posibles errores.

Además, nos ayudará a conocer como el resto de desarrolladores se organizan, analizan sus tareas y qué métodos usan para corregir sus errores. Sirviendo de referente a los demás miembros del grupo.

6. Bibliografía

Intencionadamente en blanco.