

INFORME DE ANÁLISIS

GRUPO: C1.02.11

Francisco Javier de la Prada Prados (fraprapra1@alum.us.es)

Índice

1. Introducción	2
2. Contenidos	2
2.1. Listado registro de análisis	3
2.1.1 Tareas grupales	3
2.1.2 Tareas individuales	5
3. Conclusión	7
4. Bibliografía	8

Resumen ejecutivo

En este reporte se desarrollará toda la información referente al análisis de los requisitos individuales de cada uno de los entregables que tengo que realizar

Para ello, procederé a presentar los requisitos que se me han propuesto para la entrega y ofreceré unas conclusiones detalladas del resultado de realizar un análisis de este, además de exponer las decisiones tomadas para realizar correctamente el requisito en cuestión. En caso de que el profesor realice una validación de los requisitos previo a la entrega se aportará un enlace a la validación realizada (foro de la ev o correo electrónico).

Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
15/02/2023	1.0	Creación de los documentos para el “Deliverable 1” e inicialización del proyecto	1
20/02/2023	2.0	Realización de los documentos para el “Deliverable 2”	2

1. Introducción

En primer lugar, en este documento se analizarán y explicarán cada uno de los requisitos que me han sido asignados en los entregables, tanto grupales como individuales .

Los pasos a seguir para afrontar este análisis será leer detenidamente todos los requisitos del nuevo delivery y posteriormente realizar una división entre dos tipos de requisitos:

- Los requisitos que por su simplicidad o bien por ser de los primeros en el comienzo de este proyecto no necesitaran un análisis previo.
- Los requisitos de código que por su dificultad necesitarán 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, llegado el punto en el que será necesario un análisis de todos los requisitos que me sean asignadas, así como ir aumentando el análisis sobre estos para asegurarme de realizar un buen trabajo y seguir las buenas prácticas de la asignatura.

El análisis exhaustivo de estos requisitos será muy importante ya que este me dará una idea previa de lo que debo realizar para llevar a cabo con éxito la tarea, viéndose beneficiado tanto el tiempo como el coste de la misma.

En definitiva, este documento me ayudará 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.

2. 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.

2.1. Listado registro de análisis

Voy a dividir este punto entre los requisitos grupales que tengo asignados y los propios del documento de requisitos individuales.

2.1.1 Tareas grupales

IF-09: Moneda aceptada (fraprapra1)

La configuración del sistema debe incluir los siguientes datos iniciales:

- Una moneda del sistema, que debe inicializarse en "EUR".
- Una lista de monedas aceptadas, que debe inicializarse en "EUR", "USD" y "GBP".

Análisis y toma de decisiones

Para esta tarea ha sido necesario el asesoramiento por parte de mi profesor de prácticas ya que no sabía cómo realizar esta tarea de forma satisfactoria y no encontraba ningún ejemplo parecido en las prácticas o sesiones de teoría.

Finalmente, la resolución fue que esta tarea debía resolverse modelando una entidad llamada "SystemConfiguration" o "Currency" en donde debía poner los atributos "systemCurrency" y "acceptedCurrencies" y posteriormente rellenar esos datos en el "sample-data" con las monedas aceptadas.

IF-10: Entidad Peep (fraprapra1)

Un "peep" es un mensaje publicado por cualquier persona. El sistema debe almacenar los siguientes datos sobre ellos: un momento de creación (en el pasado), un título (no vacío, más corto que 76 caracteres), un apodo (no vacío, más corto que 76 caracteres), un mensaje (no vacío, más corto que 101 caracteres), una dirección de correo electrónico opcional y un enlace opcional.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la "**Lesson 2-Data Model: Session 1-A foundation**", donde se explica claramente cómo modelar una entidad en Java, usando además las restricciones que proporciona el framework para que todos los atributos de esta sigan las reglas detalladas en el documento de requisitos grupales.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente entidades siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

IF-15: Tableros de administrador (Todos)

El sistema debe manejar los paneles de administrador con los siguientes indicadores: número total de principales con cada rol; proporción de "peeps" con una dirección de correo electrónico y un enlace; proporciones de boletines críticos y no críticos; promedio, mínimo, máximo y desviación estándar del presupuesto en las ofertas agrupadas por moneda; promedio, mínimo, máximo y desviación estándar del número de notas publicadas durante las últimas 10 semanas.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**”, donde se explica claramente cómo modelar un formulario en Java.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente formularios siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

MR-19: UML (Todos)

Elaborar un gráfico UML con las entidades del proyecto.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**” y “**Lesson 2-Data Model: Session 2-Sample Data**”, donde se pueden ver en las diapositivas numerosos ejemplos de como realizar un modelado UML correctamente utilizando la herramienta que nos proporciona el framework para dicha tarea.

2.1.2 Tareas individuales

D2-04: Rol “Lecturer”

Existe un nuevo rol específico del proyecto llamado "profesor", que tiene los siguientes datos de perfil: alma mater (no vacío, más corto que 76 caracteres), un currículum vitae (no vacío, más corto que 101 caracteres), una lista de cualificaciones (no vacío, más corto que 101 caracteres) y un enlace opcional con más información.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**”, donde se explica claramente cómo modelar un rol en Java, usando además las restricciones que proporciona el framework para que todos los atributos de esta sigan las reglas detalladas en el documentos de requisitos grupales.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente entidades siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

Por tanto, bastará con seguir las lecciones teóricas y ejemplos prácticos para realizar esta tarea.

D2-05: Entidad “Course”

Un curso agrupa varias conferencias del mismo profesor. El sistema debe almacenar los siguientes datos sobre ellos: un código (patrón "[A-Z]{1,3} [0-9]{3}", no vacío, único), un título (no vacío, más corto que 76 caracteres), un resumen (no vacío, más corto que 101 caracteres), una indicación sobre si se puede considerar un curso teórico o práctico (dependiendo de las conferencias que agrupa), un precio de venta al público (positivo o cero) y un enlace opcional con más información. Los cursos puramente teóricos deben ser rechazados por el sistema.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**”, donde se explica claramente cómo modelar una entidad en Java, usando además las restricciones que proporciona el framework para que todos los atributos de esta sigan las reglas detalladas en el documentos de requisitos grupales.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente entidades siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

Finalmente, destacar que para esta tarea ha sido necesario el asesoramiento por parte de mi profesor de prácticas ya que el atributo “CourseType” que se calcula a partir de sus “Lectures” no sabía cómo implementarlo de forma correcta. La resolución fue que dicho atributo no se calcula en la entidad, si no en la siguiente entrega donde implementaremos servicios y controladores que nos permitirán calcular dicho atributo.

D2-06: Entidad “Lecture”

Una conferencia es un documento que un profesor utiliza para transmitir conocimientos. El sistema debe almacenar los siguientes datos sobre ellos: un título (no vacío, más corto que 76 caracteres), un resumen (no vacío, más corto que 101 caracteres), un tiempo estimado de aprendizaje (en horas, positivo, no nulo), un cuerpo (no vacío, más corto que 101 caracteres), una indicación sobre si se puede considerar teórico o práctico y un enlace opcional con más información.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**”, donde se explica claramente cómo modelar una entidad en Java, usando además las restricciones que proporciona el framework para que todos los atributos de esta sigan las reglas detalladas en el documentos de requisitos grupales.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente entidades siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

D2-07: Tableros de “Lecturer”

El sistema debe manejar los paneles de profesor con los siguientes datos: número total de conferencias teóricas y prácticas; tiempo de aprendizaje promedio, desviación, mínimo y máximo de las conferencias; tiempo de aprendizaje promedio, desviación, mínimo y máximo de los cursos.

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 1-A foundation**”, donde se explica claramente cómo modelar un formulario en Java.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de como modelar correctamente formularios siguiendo los pasos correctos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

D2-07: Sample Data

Producir datos de muestra variados para probar la aplicación de manera informal. Los datos deben incluir dos cuentas de profesor con las credenciales "profesor1/profesor1" y "profesor2/profesor2".

Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 2-Data Model: Session 2-Sample Data**”, donde se explica claramente cómo introducir datos de prueba en archivos csv dentro de la carpeta “sample data” de forma que puedan ser leídos correctamente por el sistema y poder realizar tareas futuras.

Además en el workspace proporcionado por los profesores vienen ejemplos prácticos de archivos csv en los que se puede observar cómo se introducen datos de ejemplo para las entidades y roles que se hayan modelado, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo el modelado de forma correcta.

3. Conclusión

Este documento me ha sido de gran utilidad para tener claro como voy a realizar las tareas asignadas además de para dar contexto al profesor que me vaya a realizar la corrección de mis entregables. Realizar un análisis previo a la realización de los requisitos más complejos de los distintos entregables me es de gran utilidad ya que puedo trazar un plan de ruta claro y conciso sobre mis resoluciones de los requisitos.

Además queda claramente detallado cuales son las decisiones más importantes que he tomado sobre el cómo desarrollar las diferentes tareas tanto colectivas como individuales y como he solucionado los posibles errores que han ido apareciendo durante el desarrollo de las tareas.

De esta forma podré llevar a cabo un aprendizaje continuo en cuanto al análisis que realizó sobre las tareas que tengo asignadas, llevándome a realizar mejores análisis en las siguientes tareas que deba realizar, mejorando en mi rol como analista y dando mejores resultados a la hora de llevar a cabo las tareas futuras del proyecto.

4. Bibliografía

Intencionadamente en blanco.