

Informe de análisis-frajimmor2



Santos Martín, Javier
Ruiz Garrido, Javier
Jiménez Morales, Francisco Miguel
García de Tejada Delgado, José
Antonio Daniel Porcar Aragón

Información general del proyecto

PROJECT NAME		Acme-SF		
PARTICIPANTS				
Name	Email	Role	Username	Photo
Antonio Daniel Porcar Aragón	antporara@alum.us.es	Project Manager, Developer	antporara	
Francisco Miguel Jiménez Morales	frajimmor2@alum.us.es	Tester, Developer	frajimmor2	
Javier Santos Martín	javsanmar5@alum.us.es	Developer, Secretary	javsanmar5	
Javier Ruiz Garrido	javruigar2@alum.us.es	Analyst, Developer	Javiruizg	
José García de Tejada Delgado	josgardel8@alum.us.es	Operator, Developer	JoseGTD	
Stakeholders				
Francisco Miguel Jiménez Morales, Javier Ruiz Garrido, José García de Tejada Delgado, Javier Santos Martín, Antonio Daniel Porcar Aragón and José González Enriquez (the professor).				
Start date	Expected date of completion	Deliverables	Date of Document	
12/02/2024	27/05/2024	4	14/02/2024	

Tabla de contenido

Informe de análisis-frajimmor2	1
Información general del proyecto	2
Tabla de contenido	3
Resumen	4
Tabla de revisión	5
Introducción	6
Contenido	7
Conclusiones:	10
Bibliografía	11

Resumen

En este informe veremos un listado de los distintos requerimientos realizados por Francisco Miguel Jiménez Morales, frajimmor2; aportando un análisis de aquellos que lo exijan.
Los requerimientos realizados son:

	MANDATORY	SUPPLEMENTARY
GRUPAL	<ul style="list-style-type: none">Entidad banner	
INDIVIDUAL	<ul style="list-style-type: none">Entidad CodeAuditEntidad AuditRecordDashboard AuditorDatos de ejemplo de todas las entidades y roles creados	<ul style="list-style-type: none">Entidad/Rol AuditorDiagrama UML

Este documento se centrará en los cambios realizados exclusivamente por frajimmor2 en la entrega individual.

Tabla de revisión

No procede.

Introducción

Se ha realizado la creación de 2 entidades que se explicarán a continuación, un nuevo rol y un dashboard para este nuevo rol.

La estructura del documento consta de los siguientes apartados:

- **Participantes:** Participantes del proyecto
- **Indice:** Indice del documento
- **Resumen:** Reducción del contenido
- **Introducción:** Breve introducción del documento
- **Contenido:** Contenido principal del reporte
- **Conclusión:** Cierre del tema

Contenido

Lo aportado a nivel individual es lo siguiente:

- CREACIÓN DE ENTIDADES:

Code Audit: La creación de esta entidad ha tenido como problemas principales la decisión de qué tipo atributo debería tomar la propiedad Mark y colocar correctamente el patrón del código. Esta entidad tiene relación directa con el rol Auditor y la entidad Project, además está relacionada también con la entidad Audit Records.

Para el primer problema, se valoraron 2 posibles soluciones. Mark podría ser un atributo de tipo *Mark* siendo *Mark* un Enum creado con los diferentes valores que podría tomar en la propiedad, o usar un String y mediante una gran cantidad de restricciones lograr que solo pudiera tener los valores deseados.

Tras valorar y hacer un pequeño intento de cómo se haría en cada caso, se tomó la decisión de diseño de usar el Enum *Mark*. Los motivos de esta decisión fueron reducir en el futuro el trabajo en el servicio, ya que además de quitarnos las restricciones del String, sería más fácil decidir la moda usando un Enum. Como *Mark* es una moda calculada de los Audit Records asociados a la *Code Audit*, se implementó el Enum en el mismo package que la otra entidad.

Con respecto al segundo problema, con un poco de investigación en la documentación proporcionada y varios intentos, fue solucionado.

Archivos modificados:

- .\src\main\java\acme\entities\code_audit\CodeAudit.java
- .\src\main\java\acme\entities\code_audit\CodeAuditType.java

Audit Record: La creación de esta entidad ha sido mucho más fácil con respecto a la anterior. Puesto que ya se había solucionado el problema con el atributo *Mark* en *Code Audit*. El principal problema tuvo que ver con representar un intervalo de tiempo sin utilizar Duration. Esta entidad está estrechamente relacionada con *Code Audit*.

En cuanto a las decisiones de diseño, se ha optado por “sustituir” el tipo Duration utilizando dos fechas para calcular el inicio y el final del intervalo: “startInstant” y “endInstant”. El cálculo de los intervalos sería calculado en el servicio, desarrollado en próximos entregables.

Archivos modificados:

- .\src\main\java\acme\entities\audit_record\AuditRecord.java
- .\src\main\java\acme\entities\audit_record\Mark.java

Auditor: Con la experiencia de la creación previa de las entidades anteriores, crear este rol no tuvo ningún problema ni decisión de diseño. Como ya se ha comentado antes, este rol mantiene una relación estrecha con la entidad Code Audit.

Archivos modificados:

- .\src\main\java\acme\roles\Auditor.java

- CREACIÓN DE FORMS:

AuditorDashboard: Este form fue lo último en ser implementado, y careció de dificultad o problemas para ser implementado.

Archivos modificados:

- .\src\main\java\acme\forms\AuditorDashboard.java

-DATOS DE PRUEBA

Se han diseñado casos de prueba para cada una de las clases mencionadas anteriormente, con el objetivo de garantizar su funcionamiento correcto y validar su comportamiento bajo diferentes escenarios. Estos casos de prueba se han documentado en ficheros CSV.

Los datos de prueba se centran principalmente en probar el máximo de combinaciones posibles con los datos de las entidades, sin embargo, como la mayoría de datos son String, fechas o enum, no han sido demasiados ejemplos. Se han poblado todas las posibles opciones en los enum, fechas tanto recientes como antiguas y string cuya longitud se acerca a las restricciones.

Los datos de prueba han sido probados con éxito, verificando el buen funcionamiento de lo implementado.

Archivos modificados:

- .src\main\webapp\WEB-INF\resources\sample-data\auditor.csv

- .src\main\webapp\WEB-INF\resources\sample-data\code-audit.csv
- .src\main\webapp\WEB-INF\resources\sample-data\audit-record.csv

Conclusiones:

En conclusión, el proyecto fue desarrollado correctamente, abordando los problemas y decisiones que salían conforme se programaba y modelaba con el fin de optimizar tanto el trabajo como los resultados obtenidos.

Además, el uso de herramientas como Git y GitHub aportó mucho a la organización con el resto del equipo para lograr tanto los objetivos individuales como los comunes.

Para concluir cabe destacar que conforme se le iba dedicando más tiempo y trabajo al framework, la familiarización con este y la metodología de la asignatura, y la experiencia; el trabajo y sus resultados aumentaron en calidad, tanto por el tiempo necesario, como por la calidad del producto final.

Bibliografía

No procede.