

Analysis report



Grado en Ingeniería Informática - Ingeniería del Software

Diseño y Pruebas II

Curso 2023 - 2024

Código de Grupo: C1-001		
Autores	Correo	Rol
José María Baquero Rodríguez	josbaqrod@alum.us.es	Desarrollador
Pedro Pablo Santos Domínguez	pedsandom@alum.us.es	Desarrollador
Guillermo Gómez Romero	guigomrom@alum.us.es	Manager
Ángel Neria Acal	angneraca@alum.us.es	Desarrollador, operador
Manuel Vélez López	manvellop2@alum.us.es	Desarrollador

Repositorio: <<https://github.com/DP2-2023-2024-C1-001/Acme-SF-D02.git>>

Índice de Contenidos

- 1. Resumen Ejecutivo..... 3
- 2. Control de Versiones..... 5
- 3. Introducción..... 6
- 4. Contenido..... 7
- 5. Conclusión.....12
- Bibliografía..... 13

1. Resumen Ejecutivo

Objetivo:

El objetivo del informe de análisis que se ha realizado ha sido proporcionar una evaluación exhaustiva de los requisitos establecidos para el proyecto, identificando áreas clave para análisis y toma de decisiones. Se han analizado los requisitos pertinentes, se han identificado posibles problemas o desafíos, se han propuesto soluciones y se han ofrecido recomendaciones para optimizar el rendimiento y garantizar el éxito del proyecto.

Participantes:

Manager: Guillermo Gómez Romero.

Desarrolladores: José María Baquero Rodríguez, Pedro Pablo Santos Domínguez, Ángel Neria Acal, Guillermo Gómez Romero, Manuel Vélez López.

Operador: Ángel Neria Acal.

Roles y Responsabilidades llevadas a cabo:

Desarrollador : Crea la configuración de desarrollo, personaliza el proyecto inicial, implementa características y realiza testing informal.

Operador: Crea la configuración de despliegue, despliega la aplicación y mantiene la aplicación en correcto funcionamiento.

Manager: Responsable de la supervisión general del proyecto, creación de planes y tareas, así como adaptaciones y modificaciones de los mismos, también inicializa el repositorio.

Todos estos roles tienen en común la responsabilidad de escribir informes de los trabajos que realizan.

Plan de Acción:

Tras haber recibido las tareas asignadas a desarrollar, el primer paso es crear una rama para cada una de estas tareas.

Una vez creada la rama se implementa la tarea correspondiente.

Cuando la tarea se ha completado se realiza un commit and push al repositorio remoto.

Se crea una pull request y se asigna un revisor para que compruebe el código.

Una vez el revisor da el aprobado se hace un merge con la rama main del repositorio.

2.Control de Versiones

Fecha	Versión	Descripción
06/03/2024	V1.0	Creación inicial del informe de análisis

3.Introducción

El Informe de Análisis del Proyecto Acme-SF ofrece una visión integral del análisis llevado a cabo por el equipo en relación con los requisitos establecidos para este proyecto. El objetivo primordial de este informe es proporcionar una evaluación detallada de los requisitos del proyecto, destacando los análisis realizados y las decisiones adoptadas por el grupo para abordar las necesidades.

Este informe no solo proporcionará una comprensión clara de los requisitos del proyecto y las acciones tomadas para abordarlos, sino que también servirá como una guía para la toma de decisiones informadas y la mejora continua del proyecto en el futuro. A través de la transparencia y la colaboración, buscamos garantizar que el proyecto Acme-SF cumpla con las expectativas del cliente y alcance sus objetivos de manera efectiva.

4. Contenido

En esta entrega se debe implementar 5 entidades, para ello nos dividimos el trabajo, realizando cada miembro del grupo una entidad.

Tarea-003:

A claim is an opposition or contradiction posted by anyone that is made to something considered to be unjust. The system must store the following data about them: a code (pattern “C-[0-9]{4}”), not blank, unique), an instantiation moment (in the past), a heading (not blank, shorter than 76 characters), a description (not blank, shorter than 101 characters), the department to which it is addressed (not blank, shorter than 101 characters), an optional email address, and an optional link.

Tras estudiar el requisito, el desarrollador Angel Neria, decidió crear dentro del package entities, un package para cada entidad de los requisitos.

Para implementar la entidad claim, nuestro desarrollador tiene que crear una clase de java, “Claim”, en la que hay que configurar la clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “javax.validation.constraint”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

Tarea-004:

An objective allows an authenticated principal to define a goal or end towards which the actions or operations of a specific project are directed. The system must store the following data about them: an instantiation moment (in the past), a title (not blank, shorter than 76 characters), a description (not blank, shorter than 101 characters), a priority (“Low”, “Medium”, “High”), a status to indicate whether it is critical or not, a duration (must start at any moment after the instantiation moment), and an optional link with further information.

Tras estudiar el requisito, el desarrollador Pedro Pablo Santos, se decidió crear dentro del package entities, un package para cada entidad de los requisitos.

Para implementar la entidad claim, nuestro desarrollador tiene que crear una clase de java, “Objective”, en la que hay que configurar la clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “javax.validation.constraint”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

Tarea-005:

A risk, positive or negative, is a fact that has a certain impact on the development of a project and must be considered by the administrator. The system must store the following data about them: a reference (pattern “R-[0-9]{3}”), not blank, unique), an identification date (in the past), an impact (positive real number), a probability, a value (result of the multiplication of impact and probability), a description (not blank, shorter than 101 characters), and an optional link with further information.

Tras estudiar el requisito, el desarrollador Jose Maria Baquero, se decidió crear dentro del package entities, un package para cada entidad de los requisitos.

Para implementar la entidad claim, nuestro desarrollador tiene que crear una clase de java, “Risk”, en la que hay que configurar la clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “javax.validation.constraint”. Finalmente, se comprueba que todo

funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

Tarea-006:

A notice is a message posted by an authenticated principal. The system must store the following data about them: an instantiation moment (in the past), a title (not blank, shorter than 76 characters), an author (not blank, shorter than 76 characters), a message (not blank, shorter than 101 characters), an optional email address, and an optional link. The author must be computed as follows: “<username> - <surname, name>”, where “<username>” denotes the username of the principal who has posted the note and “<surname, name>” denotes his or her full name.

Tras estudiar el requisito, el desarrollador Guillermo Gómez, se decidió crear dentro del package entities, un package para cada entidad de los requisitos.

Para implementar la entidad claim, nuestro desarrollador tiene que crear una clase de java, “Notice”, en la que hay que configurar la clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “javax.validation.constraint”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

Tarea-007:

A banner allows administrators to advertise products, services, or organisations. The system must store the following data about them: an instantiation/update moment (in the past), a display period (must start at any moment after the instantiation/update moment and must last for at least one week), a link to a picture that must be stored somewhere else, a slogan (not blank, shorter than 76 characters), and a link to a target web document.

Tras estudiar el requisito, el desarrollador Manuel Vélez, se decidió crear dentro del package entities, un package para cada entidad de los requisitos.

Para implementar la entidad claim, nuestro desarrollador tiene que crear una clase de java, "Banner", en la que hay que configurar la clase con la anotación de "javax.persistence": "@Entity"; y las anotaciones de "lombok": "@Getter" y "@Setter". Además, se debe extender la clase de "AbstractEntity", que proporciona un id y una versión a la entidad. También, se debe añadir "serialVersionUID". Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de "javax.validation.constraint". Finalmente, se comprueba que todo funciona correctamente, y se realiza el "commit and push", correspondiente, y la "pull request".

Tarea-008:

Produce initial data to cold start your project; it must include an administrator account with credentials "administrator/administrator". Produce assorted sample data to test your requirements; it must include two administrator accounts with credentials "administrator1/administrator1" and "administrator2/administrator2".

Este requisito, se realizó por tres miembros del equipo, Angel Neria, Manuel Velez y Jose Maria Baquero. Para ello, se dividieron el trabajo para la creación de las tablas de las entidades, además, de añadir dos roles nuevos, administrator1 y administrator2.

La implementación se basaba en la creación de ficheros csv, en la carpeta "src/main/webapp/WEB-INF/resources/sample-data", de cada entidad, y la escritura de sus atributos en el correcto formato. También, la realización de los nuevos roles administrator, debían añadirse en el csv de "user-account.csv", y la posterior creación del csv de la entidad "Administrator" correspondiente.

Tarea-021:

The system configuration must include the following initial data:

- **A system currency, which must be initialised to “EUR”.**
- **A list of accepted currencies, which must be initialised to “EUR”, “USD”, and “GBP”.**

Esta tarea fue asignada a Guillermo Gómez, en este requisito tuvo que crear un csv con datos iniciales, en la carpeta

“src/main/webapp/WEB-INF/resources/sample-data”, además de la entidad “SystemConfiguration.java”.

En esta tarea no encontró ningún problema en su implementación, por lo que para su implementación siguió los pasos anteriormente mencionados en la creación de entidades.

Tarea-022:

The system must handle administrator dashboards with the following indicators: total number of principals with each role; ratio of notices with both an email address and a link; ratios of critical and non-critical objectives; average, minimum, maximum, and standard deviation of the value in the risks; average, minimum, maximum, and standard deviation of the number of claims posted over the last 10 weeks.

Este requisito fue estudiado por Jose Maria Baquero y Manuel Velez, y observaron que debían realizar un formulario. En principio, los atributos tenían tipos como Integer y Double, pero se cambiaron a tipos primitivos como int y double. Esto se debió a que los atributos del formulario no deben ser nulos, y como sabemos los tipos primitivos no permiten valores nulos.

Este formulario es “AdministratorDashboard”, está creado en “src/main/java/acme/forms” y todos los formularios deben extender de “AbstractForm”.

5. Conclusión

En resumen, este análisis exhaustivo ha destacado los puntos clave de los requisitos del proyecto. Se han tomado decisiones para abordar las áreas de mejora identificadas, centrándose en la optimización continua. Este informe proporciona una visión clara de la evolución de los requisitos y destaca aquellos que han requerido un análisis detallado debido a su complejidad.

Los requisitos de la segunda entrega, no han presentado grandes dificultades. Sin embargo, el tener que realizar todos los datos de ejemplo han hecho de esta entrega algo tedioso, debido a que se debía probar todos los tipo de datos posibles para cada atributo.

Bibliografía

Intencionadamente en blanco.