

Analysis Report



Grado en Ingeniería Informática - Ingeniería del Software

Diseño y Pruebas II

Curso 2023 - 2024

Código de Grupo: C1-001		
Autor	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-D03>

Índice de Contenidos

- 1. Resumen Ejecutivo..... 3
- 2. Control de Versiones..... 4
- 3. Introducción..... 5
- 4. Contenido..... 6
- 5. Conclusión.....8
- 6. Bibliografía..... 9

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
024/04/2024	V1.0	Creación inicial del documento.

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

Tarea Grupal 9 - Implementación de las operaciones en reclamaciones (Claims):

Para satisfacer los requisitos relacionados con las operaciones en reclamaciones, se ha diseñado e implementado un conjunto de funcionalidades específicas en la aplicación.

Primero, se estableció un modelo de datos para las reclamaciones, que incluye campos relevantes como identificador de reclamación, detalles de la reclamación y estado de publicación. Este modelo se integró con el repositorio correspondiente, denominado "ClaimRepository", que utiliza consultas personalizadas para acceder a la base de datos y recuperar la información necesaria sobre las reclamaciones.

En el contexto del controlador de la aplicación, se creó un controlador dedicado llamado "ClaimController" para gestionar las solicitudes relacionadas con las reclamaciones. Este controlador se encarga de enrutar las solicitudes entrantes a las operaciones adecuadas, como listar reclamaciones, mostrar detalles de reclamaciones y publicar reclamaciones.

Para permitir la publicación de reclamaciones, se implementó un flujo que requiere confirmación antes de realizar la operación, dado que las reclamaciones no pueden ser actualizadas ni eliminadas una vez publicadas. Este proceso de confirmación se integra en la lógica de negocio del

controlador, asegurando que se sigan las reglas establecidas para la gestión de reclamaciones.

Además, se desarrollaron vistas JSP (JavaServer Pages) para interactuar con las reclamaciones. Se creó una vista para listar reclamaciones y mostrar sus detalles, proporcionando una interfaz fácil de usar para los usuarios. Esta vista se diseñó teniendo en cuenta el requisito de internacionalización, asegurando que los momentos y cantidades de dinero se muestren en el formato adecuado según la configuración del idioma del usuario.

En resumen, la implementación de las operaciones en reclamaciones se ha realizado de manera integral, abordando cada aspecto del proceso desde el modelo de datos hasta la interfaz de usuario, y cumpliendo con los requisitos funcionales y no funcionales establecidos en el documento de análisis.

Tarea Grupal 10 y 11 - Implementación de las operaciones en banners:

Para abordar las operaciones relacionadas con los banners, se ha desarrollado una funcionalidad completa que permite a diferentes tipos de usuarios interactuar con los banners de manera eficiente y efectiva.

En primer lugar, se definieron los modelos de datos necesarios para representar los banners, incluyendo campos como identificador de banner, contenido, período de visualización activo y estado de publicación. Estos modelos se integraron con el repositorio correspondiente, denominado "BannerRepository", que proporciona métodos para acceder a los datos de los banners en la base de datos.

Para la visualización de banners de manera aleatoria y no intrusiva, se diseñó una lógica específica dentro del controlador principal de la aplicación. Este controlador se encarga de seleccionar un banner activo al azar y mostrarlo en cada página de la aplicación, asegurando que se cumpla con el requisito de mostrar los banners de manera discreta.

Además, se crearon vistas JSP especializadas para la gestión de banners, permitiendo a los administradores listar, mostrar detalles, crear, actualizar y eliminar banners según sea necesario. Estas vistas se desarrollaron teniendo en cuenta el requisito de internacionalización, asegurando que los momentos y cantidades de dinero se presenten correctamente en función del idioma del usuario.

En resumen, la implementación de las operaciones en banners se ha llevado a cabo de manera completa y robusta, abordando tanto la visualización aleatoria como la gestión administrativa de los banners, y cumpliendo con los

requisitos establecidos en el documento de análisis, incluida la internacionalización de datos específicos.

Tarea Grupal 26 - Implementación de las operaciones en la configuración del sistema (System Configuration):

Para cumplir con las operaciones en la configuración del sistema destinadas a los administradores, se ha implementado una serie de funcionalidades dentro de la aplicación.

Primero, se estableció un modelo de datos para almacenar la configuración del sistema, que incluye parámetros relevantes como configuración de red, ajustes de seguridad y preferencias de usuario. Este modelo se integró con el repositorio correspondiente, denominado "SystemConfigurationRepository", que proporciona métodos para acceder y actualizar la configuración del sistema en la base de datos.

En el contexto del controlador de la aplicación, se creó un controlador especializado llamado "SystemConfigurationController" para manejar las solicitudes relacionadas con la configuración del sistema. Este controlador se encarga de mostrar la configuración actual del sistema y permitir a los administradores actualizarla según sea necesario.

La actualización de la configuración del sistema se realiza a través de una interfaz de usuario intuitiva y segura, que proporciona formularios adecuados para editar los diferentes parámetros de configuración. Se implementaron mecanismos de validación para garantizar que los datos ingresados sean válidos y se respeten las restricciones de seguridad y negocio.

En resumen, la implementación de las operaciones en la configuración del sistema se ha realizado de manera sólida y completa, proporcionando a los administradores las herramientas necesarias para gestionar la configuración de la aplicación de forma eficiente y segura.

Tarea Grupal 27 y 28 - Implementación de las operaciones en objetivos (Objectives):

Para permitir a los usuarios autenticados interactuar con los objetivos del sistema, se ha desarrollado una serie de funcionalidades específicas dentro de la aplicación.

Se diseñó un modelo de datos para representar los objetivos del sistema, incluyendo campos como identificador de objetivo, descripción y estado de publicación. Este modelo se integró con el repositorio correspondiente,

denominado "ObjectiveRepository", que proporciona métodos para acceder y recuperar la información de los objetivos en la base de datos.

Se creó un controlador dedicado llamado "ObjectiveController" para manejar las solicitudes relacionadas con los objetivos del sistema. Este controlador se encarga de listar los objetivos disponibles y mostrar sus detalles a los usuarios autenticados.

Además, se implementó una funcionalidad que permite a los administradores publicar nuevos objetivos en el sistema. Este proceso de publicación requiere confirmación para garantizar que los objetivos una vez publicados no puedan ser actualizados ni eliminados.

En conjunto, la implementación de las operaciones en objetivos proporciona una manera efectiva para que los usuarios autenticados visualicen y accedan a los objetivos del sistema, mientras que los administradores pueden gestionar la publicación de nuevos objetivos de manera controlada y segura.

Tarea Grupal 29 y 30 - Implementación de las operaciones en riesgos (Risks):

Para abordar las operaciones relacionadas con los riesgos en el sistema, se ha desarrollado un conjunto de funcionalidades destinadas tanto a usuarios autenticados como a administradores.

Se estableció un modelo de datos para representar los riesgos, incluyendo campos como identificador de riesgo, descripción, nivel de riesgo y estado de publicación. Este modelo se integró con el repositorio correspondiente, llamado "RiskRepository", que proporciona métodos para acceder y administrar la información de los riesgos en la base de datos.

Se creó un controlador específico llamado "RiskController" para manejar las solicitudes relacionadas con los riesgos. Este controlador permite a los usuarios autenticados listar los riesgos en el sistema y mostrar sus detalles, mientras que los administradores tienen la capacidad adicional de publicar, actualizar y eliminar riesgos.

La publicación de riesgos sigue un flujo controlado que requiere confirmación por parte del administrador. Esto garantiza que una vez que un riesgo se publique, no pueda ser modificado ni eliminado, lo que es crucial para mantener la integridad y la consistencia de los datos.

Además de las operaciones básicas, se ha implementado una funcionalidad que permite a los administradores listar los riesgos en el sistema y mostrar

sus detalles. Esto proporciona una vista general completa de los riesgos disponibles y sus características para una mejor toma de decisiones.

En conjunto, la implementación de las operaciones en riesgos ofrece una manera eficiente y segura para que los usuarios autenticados accedan y gestionen los riesgos del sistema, mientras que los administradores tienen el control total sobre la publicación y gestión de los mismos.

Tarea Grupal 31 - Implementación de las operaciones en avisos (Notices):

Para satisfacer las operaciones relacionadas con los avisos en el sistema, se ha desarrollado una funcionalidad completa que aborda las necesidades tanto de los usuarios autenticados como de los administradores.

Se diseñó un modelo de datos para representar los avisos, que incluye campos como identificador de aviso, contenido y fecha de publicación. Este modelo se integró con el repositorio correspondiente, denominado "NoticeRepository", que proporciona métodos para acceder y administrar la información de los avisos en la base de datos.

Se creó un controlador especializado llamado "NoticeController" para manejar las solicitudes relacionadas con los avisos. Este controlador permite a los usuarios autenticados listar los avisos recientes en el sistema y mostrar sus detalles, así como publicar nuevos avisos.

La publicación de avisos requiere confirmación para garantizar que una vez publicados, no puedan ser modificados ni eliminados. Esto se implementó para mantener la coherencia y la integridad de la información de los avisos en el sistema.

Además, se desarrollaron vistas JSP (JavaServer Pages) dedicadas para la visualización de avisos, lo que permite una presentación clara y estructurada de la información para los usuarios autenticados. Estas vistas se han diseñado teniendo en cuenta el requisito de mostrar avisos no más antiguos de un mes, asegurando que solo se muestren los avisos relevantes y actualizados.

En resumen, la implementación de las operaciones en avisos proporciona una manera efectiva para que los usuarios autenticados accedan a información relevante y actualizada, mientras que los administradores tienen el control sobre la publicación y gestión de los avisos en el sistema.

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.

6. Bibliografía

Intencionadamente en blanco.