

Universidad de Sevilla
Escuela Técnica Superior de Ingeniería
Informática
Testing Report



Grado en Ingeniería Informática – Ingeniería del Software
Diseño y Pruebas II
Curso 2023 – 2024

Fecha	Versión
07/07/24	1.0

Grupo de Prácticas: C2.005	
Repositorio: https://github.com/lucantdel/Acme-SF	
Autores por orden alfabético	Correo
Lucas Antoñanzas del Villar	alvbercau@alum.us.es
Mohanad Abulatifa	jaicabher1@alum.us.es
Juan Carlos López Veiga	juacasben@alum.us.es
Álvaro Vázquez Conejo	nicherlob@alum.us.es
Manuel Orta Pérez	ronmonalb@alum.us.es

Índice

Resumen ejecutivo	2
Introducción	3
Contenido	4
Capítulo de pruebas funcionales	4
Descripción de los casos de prueba .safe	4
Descripción de los casos de prueba .hack	12
Bugs encontrados	13
Conclusiones	14
Bibliografía	14

Resumen ejecutivo

Este informe de pruebas documenta los resultados obtenidos a través de las pruebas funcionales y de rendimiento realizadas en nuestro proyecto. Las pruebas se dividen en dos capítulos principales: pruebas funcionales y pruebas de seguridad.

Capítulo de Pruebas Funcionales

Este capítulo detalla los casos de prueba implementados para la entidad "Contract" y "Progress Log", organizados según las características específicas del sistema. Cada caso de prueba incluye:

- Descripción del Caso de Prueba: Un resumen del objetivo del caso de prueba.
- Casos Positivos y Negativos: Evaluaciones de cómo se comporta el sistema bajo diferentes condiciones.

Capítulo de Pruebas de Seguridad

Este capítulo se enfoca en los intentos de acceder a funcionalidades del sistema sin los permisos adecuados, probando la robustez de las restricciones de acceso y validaciones de seguridad.

Tabla de revisiones

Número de revisión	Fecha	Descripción
1	07/07/2024	Creación y realización del documento

Introducción

El propósito de este informe es proporcionar una documentación detallada de los resultados obtenidos a través de las pruebas funcionales y de seguridad realizadas en nuestro proyecto. Estas pruebas son cruciales en el ciclo de desarrollo de software para garantizar la calidad, seguridad y fiabilidad del producto final.

Pruebas Funcionales

En el capítulo de pruebas funcionales, se describen en detalle los casos de prueba implementados para las entidades "Contract" y "Progress Log". Cada caso de prueba incluye una descripción y evaluación de su efectividad en la detección de errores. Este enfoque asegura que todas las funcionalidades críticas del sistema han sido evaluadas exhaustivamente.

Pruebas de Seguridad

El capítulo de pruebas de seguridad se centra en verificar que el sistema implementa correctamente las restricciones de acceso y que solo los usuarios autorizados pueden realizar ciertas acciones. Esto incluye pruebas de acceso con roles incorrectos y pruebas de acceso con cuentas de usuario no autorizadas.

Contenido

Capítulo de pruebas funcionales

Descripción de los casos de prueba .safe

1- Entidad Contract.

ClientContractListMineService.java

- Se accede al listado de contratos iniciando sesión desde los diferentes clientes registrados y se registra al administrador como cliente ("become client") para acceder al listado.

ClientContractShowService.java

- Se accede a varios formularios de contratos iniciando sesión desde los diferentes clientes registrados.

ClientContractCreateService.java

Atributo	Casos positivos	Casos negativos
code	<ul style="list-style-type: none">- Valores límite inferior (AA-AA-0000)- Valores límite superior (ZZ-ZZ-9999)- Formato válido cualquier valor	<ul style="list-style-type: none">- Vacío- Formato incorrecto- Duplicado

Provider Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Customer Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Goals	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (100) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (101)
Budget	<ul style="list-style-type: none"> - Divisas aceptadas por el sistema - Cantidades positivas con decimales - Cantidades positivas sin decimales - Valores límite superior (1 000 000) 	<ul style="list-style-type: none"> - Vacío - Divisas no aceptadas por el sistema - Cantidades negativas - Cantidades negativas + divisas no aceptadas - Valores que superan límite superior (1 000 000)
Draft Mode	<ul style="list-style-type: none"> - True o False 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto

ClientUpdateUpdateService.java

Atributo	Casos positivos	Casos negativos
code	<ul style="list-style-type: none"> - Valores límite inferior (AA-AA-0000) - Valores límite superior (ZZ-ZZ-9999) - Formato válido cualquier valor 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto - Duplicado
Provider Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Customer Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Goals	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (100) 	<ul style="list-style-type: none"> - Vacío Valores que superan límite superior (101)

Budget	<ul style="list-style-type: none"> - Divisas aceptadas por el sistema - Cantidades positivas con decimales - Cantidades positivas sin decimales - Valores límite superior (1 000 000) - 	<ul style="list-style-type: none"> - Vacío - Divisas no aceptadas por el sistema - Cantidades negativas - Cantidades negativas + divisas no aceptadas - Valores que superan límite superior (1 000 000)
Draft Mode	<ul style="list-style-type: none"> - True o False 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto

ClientContractPublishService.java

Atributo	Casos positivos	Casos negativos
code	<ul style="list-style-type: none"> - Valores límite inferior (AA-AA-0000) - Valores límite superior (ZZ-ZZ-9999) - Formato válido cualquier valor 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto - Duplicado
Provider Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)

Customer Name	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Goals	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (100) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (101)
Budget	<ul style="list-style-type: none"> - Divisas aceptadas por el sistema - Cantidades positivas con decimales - Cantidades positivas sin decimales - Valores límite superior (1 000 000) 	<ul style="list-style-type: none"> - Vacío - Divisas no aceptadas por el sistema - Cantidades negativas - Cantidades negativas + divisas no aceptadas - Valores que superan límite superior (1 000 000)
Draft Mode	<ul style="list-style-type: none"> - True o False 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto

Además, como casos negativos se ha intentado publicar un contrato:

- Contrato con divisa distinta al proyecto que este el contrato asociado

ClientContractDeleteService.java

- Se borran proyectos iniciando sesión desde los diferentes clientes registrados.

2- Entidad Progress Log

ClientProgressLogListService.java

- Se accede al listado de asignaciones iniciando sesión desde los diferentes clientes registrados y se registra al administrador como cliente ("become client") para acceder al listado.

ClientProgressLogShowService.java

- Se accede a varios formularios de asignaciones iniciando sesión desde los diferentes clientes registrados.

ClientProgressLogCreateService.java

Atributo	Casos positivos	Casos negativos
Record Id	<ul style="list-style-type: none">- Valores límite inferior (AA-AA-0000)- Valores límite superior (ZZ-ZZ-9999)- Formato válido cualquier valor	<ul style="list-style-type: none">- Vacío- Formato incorrecto- Duplicado
Completeness	<ul style="list-style-type: none">- Valores límite inferior (0)- Valores límite superior (100)	<ul style="list-style-type: none">- Vacío- Valores que superan límite superior (101)- Cantidades negativas

Comment	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (100) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (101)
Responsible Person	<ul style="list-style-type: none"> - Valores límite inferior (1) - Valores límite superior (75) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (76)
Draft Mode	<ul style="list-style-type: none"> - True o False 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto

ClientProgressLogUpdateService.java

Atributo	Casos positivos	Casos negativos
Record Id	<ul style="list-style-type: none"> - Valores límite inferior (AA-AA-0000) - Valores límite superior (ZZ-ZZ-9999) - Formato válido cualquier valor 	<ul style="list-style-type: none"> - Vacío - Formato incorrecto - Duplicado
Completeness	<ul style="list-style-type: none"> - Valores límite inferior (0) - Valores límite superior (100) 	<ul style="list-style-type: none"> - Vacío - Valores que superan límite superior (101) - Cantidades negativas

Comment	<ul style="list-style-type: none">- Valores límite inferior (1)- Valores límite superior (100)	<ul style="list-style-type: none">- Vacío- Valores que superan límite superior (101)
Responsible Person	<ul style="list-style-type: none">- Valores límite inferior (1)- Valores límite superior (75)	<ul style="list-style-type: none">- Vacío- Valores que superan límite superior (76)
Draft Mode	<ul style="list-style-type: none">- True o False	<ul style="list-style-type: none">- Vacío- Formato incorrecto

ClientProgressLogDeleteService.java

- Se borran historias de usuario iniciando sesión desde los diferentes clientes registrados.

Descripción de los casos de prueba .hack

1- Entidad Contract

ClientContractListMineService.java

- Wrong role: se intenta acceder al listado de contratos como usuario sin identificar, administrador y otro rol cualquiera.

ClientContractShowService.java

- Wrong role: se intenta acceder al formulario de un contrato como usuario sin identificar, administrador y otro rol cualquiera.
- Wrong user account: se intenta acceder al formulario de un contrato con un cliente diferente al propietario de este.

ClientContractCreateService.java

- Wrong role: se intenta crear un contrato como usuario sin identificar, administrador y otro rol cualquiera.

ClientContractUpdateService.java

- Wrong role: se intenta actualizar un contrato como usuario sin identificar, administrador y otro rol cualquiera.
- Wrong user account: se intenta actualizar un contrato en estado borrador con un cliente diferente al propietario de este utilizando la url.
- Wrong action: se intenta actualizar un contrato en estado publicado con el cliente propietario de este utilizando la url.

ClientContractPublishService.java

- Wrong role: se intenta publicar un contrato como usuario sin identificar, administrador y otro rol cualquiera..

ClientContractDeleteService.java

- Wrong role: se intenta eliminar un proyecto como usuario sin identificar, administrador y otro rol cualquiera.

2- Entidad Progress Log

ProgressLogListService.java

- Wrong role: se intenta acceder al listado de Progress Log como usuario sin identificar, administrador y otro rol cualquiera.

ProgressLogShowService.java

- Wrong role: se intenta acceder al formulario de un Progress Log como usuario sin identificar, administrador y otro rol cualquiera.
- Wrong user account: se intenta acceder al formulario de una asignación con un cliente diferente al propietario de este.

ProgressLogCreateService.java

- Wrong role: se intenta crear un ProgressLog como usuario sin identificar, administrador y otro rol cualquiera.

ProgressLogUpdateService.java

- Wrong role: se intenta actualizar un ProgressLog como usuario sin identificar, administrador y otro rol cualquiera.
- Wrong user account: se intenta actualizar un ProgressLog en estado borrador con un cliente diferente al propietario de este.
- Wrong action: se intenta actualizar un ProgressLog en estado publicado con el cliente propietario de este.

ProgressLogPublishService.java

- Wrong role: se intenta publicar un ProgressLog como usuario sin identificar, administrador y otro rol cualquiera.
- Wrong user account: se intenta publicar un ProgressLog en estado borrador con un cliente diferente al propietario de este.
- Wrong action: se intenta publicar un ProgressLog en estado publicado con el cliente propietario de este.

ProgressLogDeleteService.java

- Wrong role: se intenta eliminar un ProgressLog como usuario sin identificar, administrador y otro rol cualquiera.

Bugs encontrados

Tras la realización de todos los tests anteriormente mencionados se encontraron los siguientes bugs:

Bug	Solución
En los campos "code" "recordId" de las entidades contract y progressLog, introducir un código, recordId con longitud mayor que 255.	Añadir @Length(max = 255) a los atributos en las entidades mencionadas
Se permitía asociar una divisa al contrato diferente a la del proyecto al que está asociado	Implementar validación para asegurar que la divisa del contrato sea la misma que la del proyecto asociado
Se podía acceder a un contrato sin tener la autorización correspondiente.	Añadir validación para asegurar que solo los usuarios autorizados puedan acceder a los contratos

Se podía modificar la fecha de creación de un contrato	Hacer que la fecha de creación se asigne automáticamente y no sea modificable
Se podía modificar la fecha de registro de un "progress log"	Hacer que la fecha de registro se asigne automáticamente y no sea modificable
El atributo "completeness" al crear un "progress log" permitía ser nulo	Añadir la restricción para que el atributo "completeness" no pueda ser nulo
Se permitía asociar un contrato a un proyecto estando en draft mode	Permitir que los contratos se asignen únicamente a proyectos ya publicados

Conclusiones

El análisis de las pruebas funcionales y de seguridad realizadas en nuestro proyecto ha mostrado resultados positivos en términos de eficiencia y fiabilidad del sistema.

Pruebas Funcionales

Las pruebas funcionales han cubierto exhaustivamente todas las funcionalidades críticas del sistema, documentando tanto los casos positivos como negativos, lo cual proporciona una visión clara de la robustez del sistema.

Pruebas de Seguridad

Las pruebas de seguridad han confirmado que el sistema implementa correctamente las restricciones de acceso, asegurando que solo los usuarios autorizados pueden acceder a funcionalidades específicas.

En resumen, el sistema ha demostrado ser robusto y seguro, cumpliendo con los requisitos esperados y proporcionando una base sólida para su despliegue en producción.

Bibliografía

Intencionalmente en blanco.