

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Informe de test



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2022 – 2023

Fecha	Versión
26 / 05 / 2023	V4.0

Grupo: C1.02.05	
Repositorio: https://github.com/marvicmar/Acme-L3-D04.git	
Miembros	Correo
Pérez Vázquez, Alejandro	alepervaz@alum.us.es

Tutor: Soria Morillo, Luís Miguel

Tabla de Contenidos

Tabla de Revisión	3
Resumen	4
Introducción	5
SonarLint	8
Advertencias ignoradas	8
Tiempos de la aplicación	9
Conclusiones	11
Bibliografía	12

Tabla de Revisión

Nº de revisión	Fecha	Descripción	Sprint
4.0	26 / 05 / 2023	Se describen algunos aspectos del proyecto relativos a los test y la calidad del código.	4

Resumen

Durante este cuatrimestre, en la asignatura Diseño y Pruebas II, se llevaron a cabo varias entregas de proyectos en equipo. El objetivo de este informe es presentar los requisitos solicitados por el profesorado y las soluciones propuestas por los alumnos del grupo de trabajo, en particular nos enfocaremos en el proyecto Acme-L3. En este informe se prestará especial atención a las decisiones tomadas por el grupo C1.02.05 y el estudiante Alejandro Santiago Félix.

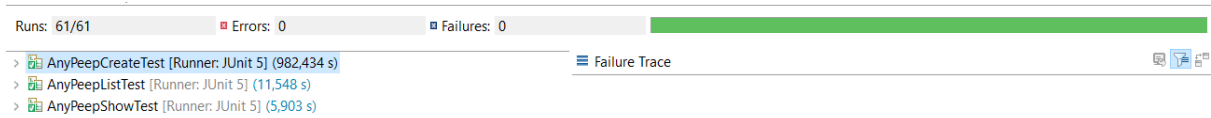
Es importante mencionar que, en aquellos casos en los que los requisitos y/o las soluciones no presenten dificultades significativas o ambigüedades, se proporcionará una descripción más breve o se omitirá, según corresponda a la situación.

Introducción

En este informe se proporcionan detalles sobre los resultados obtenidos utilizando SonarLint, así como los tiempos de ejecución tanto del grupo de trabajo C1.02.05 como del estudiante Alejandro Santiago Félix, identificado como "Estudiante #4" en GitHub y EV. El propósito principal de este informe es explicar las razones por las cuales algunos problemas de código identificados por SonarLint no han sido corregidos, a pesar de las sugerencias realizadas, y también se busca documentar el rendimiento de la aplicación.

Ejecución de test

Si todos los tests se ejecutan, como se puede observar en la imagen adjunta, se espera que se ejecuten correctamente sin ningún problema.



Según se muestra en la imagen adjunta, el porcentaje de código que debe ser cubierto es aproximadamente del 23%.

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
> Hello-World-22.0	22,9 %	5.042	16.947	21.989
> Acme-Framework-23.4.1	47,8 %	12.363	13.520	25.883

SonarLint

El objetivo de SonarLint es evitar las malas prácticas a la hora de escribir código, de esta forma se garantiza un mínimo de calidad de código.

Advertencias ignoradas

Advertencia	Razón
Replace this assert with a proper check.	Debido a las pautas impuestas por la asignatura se mantienen los asserts.
Rename this package name to match the regular expression '^[a-z_]+(\\.[a-z_][a-z0-9_]*)*\$'.	Debido a las pautas impuestas por la asignatura se nombrarán los paquetes siguiendo la nomenclatura camelCase..
Define a constant instead of duplicating this literal "name" n times.	Consideramos que al tratarse de una propiedad de cada tipo y como se repiten pocas veces no se declarará ninguna variable con ese fin, ya que lo consideramos redundante.

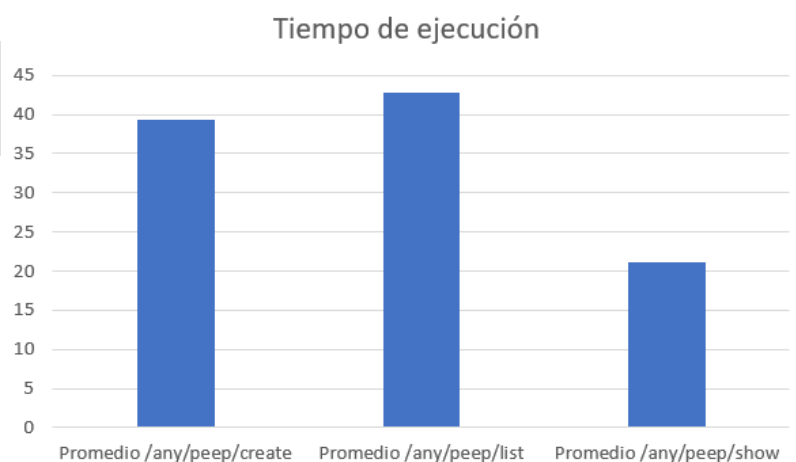
Description	Resource	Path	Location	Type
▼ ⚠ Warnings (6 items)				
🔔 The import acme.entities.banner.Banner is never used	AdministratorOff...	/Hello-World-22.0/sr...	line 4	Java Probl...
🔔 The import acme.framework.components.accounts.Administrator is ne	AuthenticatedOff...	/Hello-World-22.0/sr...	line 13	Java Probl...
🔔 The value of the local variable choice is not used	AuditingRecordS...	/Hello-World-22.0/sr...	line 92	Java Probl...
🔔 The value of the local variable choice is not used	AuditorAuditingR...	/Hello-World-22.0/sr...	line 91	Java Probl...
🔔 The value of the local variable tutorial is not used	AssistantSessionC...	/Hello-World-22.0/sr...	line 49	Java Probl...
🔔 The value of the local variable userAccount is not used	TutorialShowServ...	/Hello-World-22.0/sr...	line 43	Java Probl...
▼ ⓘ Infos (8 items)				
🔔 Define a constant instead of duplicating this literal "email" 3 times.	AnyPeepCreateSe...	Hello-World-22.0/src...	line 34	SonarLint ...
🔔 Define a constant instead of duplicating this literal "message" 3 times	AnyPeepCreateSe...	Hello-World-22.0/src...	line 34	SonarLint ...
🔔 Define a constant instead of duplicating this literal "title" 3 times.	AnyPeepCreateSe...	Hello-World-22.0/src...	line 34	SonarLint ...
🔔 Override the "equals" method in this class.	Peep.java	Hello-World-22.0/src...	line 24	SonarLint ...
🔔 Remove this commented out code.	pom.xml	Hello-World-22.0	line 68	SonarLint ...
🔔 Replace this assert with a proper check.	AnyPeepCreateSe...	Hello-World-22.0/src...	line 83	SonarLint ...
🔔 Replace this assert with a proper check.	AnyPeepCreateSe...	Hello-World-22.0/src...	line 92	SonarLint ...
🔔 Replace this assert with a proper check.	AnyPeepCreateSe...	Hello-World-22.0/src...	line 110	SonarLint ...

Tiempos de la aplicación

Se muestran distintas gráficas referentes a los tiempos de ejecución de la aplicación junto con sus casos de prueba, .

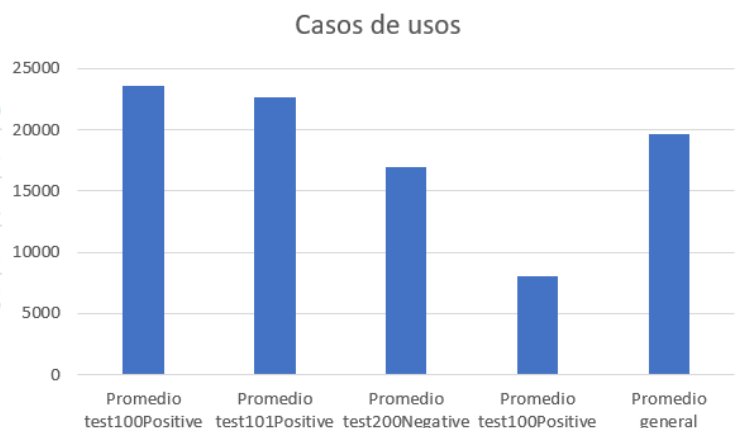
La primera gráfica muestra los tiempos promedio de ejecución de diferentes funcionalidades de las entidades "Practicum" y "SessionPracticum" en el equipo del Estudiante 4. Estos datos se pueden encontrar en la tabla correspondiente que se encuentra a la izquierda de la gráfica.

Promedio /any/peep/create	39,3921569
Promedio /any/peep/list	42,7226277
Promedio /any/peep/show	21,0697674
Promedio general	38,2163121



En la segunda gráfica se muestran los tiempos promedio de ejecución de los diferentes tipos de pruebas (positivas, negativas y de hacking) realizadas en las entidades "Peep" relacionadas con el estudiante en cuestión. Estos tiempos medios proporcionan información sobre la eficiencia y la duración de las pruebas realizadas en estas entidades.

Promedio test100Positive	23601,6216
Promedio test101Positive	22685
Promedio test200Negative	16929,7692
Promedio test100Positive	8055,1
Promedio general	19616,1148



En esta captura se presentan datos relacionados con el tiempo de ejecución de las características o funcionalidades del sistema, junto con los intervalos de confianza. Los

intervalos de confianza son medidas estadísticas que nos indican la precisión de los tiempos de ejecución estimados. Estos datos nos brindan información sobre la eficiencia y la estabilidad de las diferentes características del sistema, permitiéndonos evaluar su rendimiento en términos de tiempo de ejecución.

Datos					
Media	38,2163121				
Error típico	1,62505203		Intervalo(ms)	35,0174912	41,4151329
Mediana	36,5		Intervalo(s)	0,03501749	0,04141513
Moda	15				
Desviación estándar	27,289264				
Varianza de la muestra	744,703932				
Curtosis	59,1876572				
Coeficiente de asimetría	6,23482319				
Rango	323				
Mínimo	13				
Máximo	336				
Suma	10777				
Cuenta	282				
Nivel de confianza(95,0%	3,19882081				

Conclusiones

La correcta redacción de un informe de este tipo representa algo fundamental para el correcto seguimiento de proyectos tecnológicos, no solo para poder comprender qué problema ha sido necesario resolver, sino también para conocer la perspectiva de cada uno de los implicados, así como los problemas con los que se han encontrado en su realización.

Hemos sufrido algunos inconvenientes debido a la incompatibilidad del framework con equipos mac y linux. A su vez al necesitar de equipos windows hemos tenido problemas, relacionados con el tiempo que lleva lanzar todos los test debido a la potencia de los equipos.

Bibliografía

Intencionalmente en blanco.