Práctica 2

Pruebas de sistema y análisis estático de código

Objetivo

Se desean implementar ciertos controles de calidad de la aplicación ***Nitflex***, que gestiona películas. Esta aplicación ofrece un interfaz web para la gestión de películas y sus reseñas. Se proporciona el código de dicha aplicación en el aula virtual, que será el mismo que en la Práctica 1. Se recomienda tomar como punto de partida la solución generada de la Práctica 1.

Tarea A: Implementación de pruebas de sistema con Selenium

El grupo de alumnos deberá implementar las pruebas automáticas descritas a continuación:

- **Pruebas de la interfaz web de la aplicación con Selenium (5 puntos)**
  Se desea comprobar que**:**
  - Cuando se da de alta una nueva película (sin incluir la imagen), esperamos que la película creada aparezca en la pantalla resultante
  - Cuando se da de alta una nueva película sin título, esperamos que se muestre un mensaje de error y que no aparece esa película en la página principal

Consideraciones de la Tarea A

Se deberán tener en cuenta las siguientes consideraciones para la realización de la práctica:

- Las pruebas deben ser independientes entre sí: no depender de información que otras pruebas hayan creado o eliminado ni tampoco de los recursos creados en DatabaseInitializer.
- El profesor ejecutará todos los test a la vez (utilizando **mvn test**) y estos deberán pasar. Si una prueba no pasa (su resultado es fallo) no puntuará.
- Se valorará la modularización de las pruebas en paquetes o clases diferentes, dado que son de diferente naturaleza.
- Si una funcionalidad debe lanzar un error, se deben hacer las comprobaciones pertinentes sobre qué error se lanza, comprobando además el mensaje de error esperado.
- Puede modificarse el HTML proporcionado para añadir únicamente parámetros (como id 's) a las etiquetas HTML que faciliten las pruebas de Selenium.

Tarea B: Análisis estático de código con Sonar + JaCoCo

Se debe analizar el código usando Sonarqube y analizar los issues reportados por la herramienta:

- Cada issue deberá ser reportado y corregido. Para ello, se usará la información reportada por la herramienta sobre el issue como punto de partida para buscar una solución a la misma.
- Si se considera que un issue es un "falso positivo" porque el código es correcto no se corregirán y se marcarán en Sonarqube como "Resolve as false positive".
- Además del análisis estático, se deberá recoger la cobertura actual de las pruebas mediante Jacoco (se pueden incluir las pruebas unitarias y de integración de la anterior práctica para aumentar la cobertura)

Consideraciones de la Tarea B:

- Pueden ignorarse los issues de la categoría "Info"
- Esta parte se evaluará esencialmente utilizando la memoria que realizará el alumno (ver más abajo)

Formato de entrega

La práctica se entregará por el aula virtual teniendo en cuenta los siguientes aspectos:

- **No se garantiza que se respondan dudas de la práctica 48 horas antes de su entrega**.
- Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas sólo será entregada por uno de los alumnos.
- La práctica se entregará como un fichero .zip del proyecto Maven. El nombre del fichero .zip será el usuario URJC del alumno. En caso de dos alumnos, el nombre del zip será el usuario URJC separado por guion (p.perezf-z.gonzalez.zip)

Además del código fuente, se deberá elaborar una memoria explicativa del mismo en **formato PDF**.

- Deberá guardarse en la carpeta raíz del proyecto.
- La portada deberá contener el nombre de la práctica, la asignatura, el curso académico y el nombre del alumno o alumnos.
- La memoria deberá contener:
    - Una captura de pantalla del dashboard (Overview) dónde se vean las principales métricas: Open Issues (Security, Reliability y Mantainability), Coverage … tras un primer análisis (antes de corregir las issues)
    - Por cada issue encontrada se clasificará si es real o un falso positivo:
        - Si es **real:**
            - Reportar la issue (captura de pantalla)
            - Explicar con tus palabras en que consiste la issue
            - Corregirla (en el código) y explicar la solución (mostrando el fragmento de código)
        - Si es un **falso positivo**
            - Reportar la issue (captura de pantalla)
            - Razonar con tus palabras por qué no es una issue (o por qué Sonar no debería considerarla una issue)
    - Una vez solucionadas todas la issues, se deberá mostrar de nuevo el dashboard (Overview) en su pestaña “New code”
- La calidad de la memoria será evaluada
    - Seguir un formato adecuado (utilizar índices, secciones …)
    - Acotar las capturas de pantalla para que sea visible lo verdaderamente relevante
    - Al mostrar un fragmento de código: utilizar una captura de pantalla o texto enriquecido (highlight)