|  |  |  |
| --- | --- | --- |
| Sprint: | Scrum 4 | |

|  |  |  |
| --- | --- | --- |
| Título | Sonarqube al módulo de auditoría | |
| Versión Nº | Rama gonzalo 12-11-2018 | |
| Actores | Docentes, Director. | |
| Descripción | El objetivo de este módulo es la administración de auditorías cuando se presentan problemas con los profesores. | |
| Inicializador | Reporte automática centrada en el profesor que no contesta consultas.  Reporte de parte del alumno indicando que el profesor no atiende consultas. | |
| Escenario principal | Pasos | Acción |
|  | 1. | Se da inicio a SonarQube mediante la ejecución del archivo StartSonar ubicado en la carpeta de SonarQube(el cual se debe instalar en el sistema). |
|  | 2. | Se ingresa el archivo con extensión .properties (con su respectiva configuración para el test) en la carpeta principal del código a analizar. |
|  | 3. | Se ejecuta la ventana de comandos dentro de la carpeta principal y se ejecuta el comando “sonar-scanner” ubicado en la carpeta SonarScanner. |
| Requisitos especiales | 1. Contar con una plataforma compatible con la aplicación/programa para un inicio correcto. 2. Contar con una cuenta para el correcto inicio de sesión en la aplicación/programa. 3. Contar con conexión a internet. | |
| Suposiciones | 1. El usuario conoce sus credencial para ingresar a la aplicación | |
| Pre-Condiciones | 1. El usuario debe estar registrado en el sistema. 2. El usuario debe haber ingresado a algún módulo de información en específico. | |
| Post-Condiciones | 1. El usuario ingresa correctamente al programa | |
| Interfaz de usuario | 1. Se mostrará la información correspondiente al usuario. | |
| Resultado esperado | 1. Se espera que el código pase el test y no presente bugs, vulnerabilidades o code smells significativos. | |

|  |  |  |
| --- | --- | --- |
| Resultado Obtenido | Views.py presenta 0 bugs, 0 problemas de seguridad y 2 smells donde nos recomienda refactorizar la función de la línea 13así como eliminar los comentarios de la línea 9. | |