Código duplicado

- Nos centramos en localizar bloques de código duplicado puesto que dificulta el mantenimiento del mismo e indica que esa parte debe ser refactorizada

Seguir estándares de código y buenas prácticas

- Mediante análisis estático localizamos partes del código que cumplen con las buenas prácticas definidas para el lenguaje desarrollo o supongan errores potenciales (variables no inicializadas correctamente, referencias nulas, bucles infinitos, condicionales erróneos, etc.)
- Mantener la uniformidad del repositorio de código en cuanto a formato y estructura

Complejidad código

- Análisis de la complejidad ciclomática de los métodos y clases
- Ficheros dependientes entre paquetes o módulos

Diseño - Cohesión y Acoplamiento

- Evitar dependencias de paquetes o módulos cíclicas
- LCOM4 - Medir las relaciones entre los métodos y variables de una clase. Valores altos indican que la clase debería ser dividida en varias

Pruebas unitarias

- Número de pruebas ejecutadas y su % de éxito
- Cobertura de código, centrándonos en los siguiente aspectos
    - Líneas no cubiertas
    - % de cobertura en líneas añadidas en el último commit. Esto indica si estamos nuevos test a medida que añadimos nueva funcionalidad

Documentación y comentarios

- Asegurarnos que existen comentarios en el código
- Comprobar que los métodos de las APIs están correctamente documentamos

---

Fuentes:
https://es.slideshare.net/keheliya/sonar-metrics
https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Size
http://www.aivosto.com/project/help/pm-oo-cohesion.html#LCOM4