**Definiciones de métricas**

**Complejidad**

**Complejidad (complexity)**  
Es la Complejidad Ciclomática calculada en base al número de caminos a través del código. Siempre que el flujo de control de una función se divide, el contador de complejidad se incrementa en uno. Cada función tiene una complejidad mínima de 1. Este cálculo varía ligeramente según el idioma porque las palabras clave y las funcionalidades lo hacen.

**Complejidad cognitiva** (cognitive\_complexity)  
Qué tan difícil es comprender el flujo de control del código. Consulte [el Libro blanco de complejidad cognitiva](https://www.sonarsource.com/resources/white-papers/cognitive-complexity.html)  para obtener una descripción completa del modelo matemático aplicado para calcular esta medida.

**Duplicaciones**

**Bloques duplicados** (duplicated\_blocks)  
Número de bloques de líneas duplicados.

**Archivos duplicados** (duplicated\_files)  
Número de archivos involucrados en duplicaciones.

**Líneas duplicadas** (duplicated\_lines)  
Número de líneas involucradas en duplicaciones.

**Líneas duplicadas (%)** ( duplicated\_lines\_density)  
Se calcula mediante: duplicated\_lines/ lines\* 100

**Cuestiones**

**Problemas nuevos** (new\_violations)  
Número de problemas planteados por primera vez sobre el nuevo código.

**Problemas nuevos** **xxx** (new\_xxx\_violations)  
Número de asuntos de la gravedad especifica planteado por primera vez el nuevo código, donde xxx es uno de: blocker, critical, major, minor, info.

**Problemas** (violations)  
Recuento total de problemas en todos los estados.

**Problemas xxx** (xxx\_violations)  
Recuento total de los problemas de la gravedad especificado, donde XXX es uno de: blocker, critical, major, minor, info.

**Problemas de falsos positivos** (false\_positive\_issues)  
Recuento total de problemas marcados como falso positivo

**Problemas abiertos** (open\_issues)  
Recuento total de problemas en el estado Abierto.

**Problemas confirmados** (confirmed\_issues)  
Recuento total de problemas en el estado Confirmado.

**Problemas reabiertos** (reopened\_issues)  
Recuento total de problemas en el estado Reabierto

**Mantenibilidad**

**Code Smells** (code\_smells)  
Recuento total de problemas de Code Smell.

**News Code Smells** (new\_code\_smells)  
Recuento total de problemas de olor de código que se plantearon por primera vez en Código nuevo.

**Calificación de** **mantenibilidad** (sqale\_rating)  
Calificación otorgada a su proyecto en relación con el valor de su índice de deuda técnica. La cuadrícula de clasificación de mantenibilidad predeterminada es:

A = 0-0,05, B = 0,06-0,1, C = 0,11-0,20, D = 0,21-0,5, E = 0,51-1

La escala de calificación de mantenibilidad se puede establecer alternativamente diciendo que si el costo de remediación pendiente es:

* <= 5% del tiempo que ya ha entrado en la aplicación, la calificación es A
* entre el 6 y el 10% la calificación es una B
* entre el 11 y el 20% la calificación es una C
* entre el 21 y el 50% la calificación es D
* cualquier cosa por encima del 50% es una E

**Deuda técnica** (sqale\_index)  
Esfuerzo para corregir todos los olores de código. La medida se almacena en minutos en la base de datos. Se asume un día de 8 horas cuando los valores se muestran en días.

**Deuda técnica en código nuevo** (new\_technical\_debt)  
Esfuerzo para corregir todos los olores de código que se generaron por primera vez en código nuevo.

**Ratio de deuda técnica** (sqale\_debt\_ratio)  
Relación entre el coste de desarrollo del software y el coste de reparación. La fórmula del coeficiente de endeudamiento técnico es:  
Remediation cost / Development cost  
que se puede reexpresar como:  
Remediation cost / (Cost to develop 1 line of code \* Number of lines of code)  
El valor del costo para desarrollar una línea de código es de 0.06 días.

**Relación de deuda técnica en el nuevo código** (new\_sqale\_debt\_ratio)  
Relación entre el costo de desarrollar el código modificado en el nuevo código y el costo de las cuestiones vinculadas a él.

**Puertas de calidad**

**Quality Gate Status** (alert\_status)  
Estado del Quality Gate asociado a su proyecto. Los valores posibles son: ERROR, OK WARN valor se ha eliminado desde 7,6.

**Detalles de Quality Gate** (quality\_gate\_details)  
Para todas las condiciones de su Quality Gate, usted sabe qué condición está fallando y cuál no.

**Fiabilidad**

**Errores** (bugs)  
Número de problemas de errores.

**Errores nuevos** (new\_bugs)  
Número de problemas de errores nuevos.

**Índice de confiabilidad** (reliability\_rating)  
A = 0 errores  
B = al menos 1 error menor  
C = al menos 1 error mayor  
D = al menos 1 error crítico  
E = al menos 1 error bloqueador

Esfuerzo de corrección de **confiabilidad** (reliability\_remediation\_effort)  
Esfuerzo para solucionar todos los problemas de errores. La medida se almacena en minutos en la base de datos. Se asume un día de 8 horas cuando los valores se muestran en días.

**Esfuerzo de corrección de confiabilidad en el nuevo código** (new\_reliability\_remediation\_effort)  
Igual que el *esfuerzo de corrección de confiabilidad,* pero en el código modificado en Código nuevo.

**Seguridad**

**Vulnerabilidades** (vulnerabilities)  
Número de problemas de vulnerabilidad.

**Vulnerabilidades en código nuevo** (new\_vulnerabilities)  
Número de nuevos problemas de vulnerabilidad.

**Calificación de seguridad** (security\_rating)  
A = 0 vulnerabilidades  
B = al menos 1 vulnerabilidad menor  
C = al menos 1 vulnerabilidad mayor  
D = al menos 1 vulnerabilidad crítica  
E = al menos 1 vulnerabilidad de bloqueador

**Esfuerzo de corrección de** **seguridad** (security\_remediation\_effort)  
Esfuerzo para solucionar todos los problemas de vulnerabilidad. La medida se almacena en minutos en la base de datos. Se asume un día de 8 horas cuando los valores se muestran en días.

**Esfuerzo de corrección de seguridad en código nuevo** (new\_security\_remediation\_effort)  
Igual que *esfuerzo de corrección de seguridad,* pero en el código cambiado en Código nuevo.

**Security Hotspots** (security\_hotspots) Número de Security Hotspots

**Puntos de acceso de seguridad en código nuevo** (new\_security\_hotspots) Número de puntos de acceso de seguridad nuevos en código nuevo.

**Calificación de revisión de seguridad** (security\_review\_rating)

La calificación de revisión de seguridad es una calificación de letra basada en el porcentaje de puntos de acceso de seguridad revisados ​​(fijos o seguros).

A => = 80%  
B => = 70% y <80%  
C => = 50% y <70%  
D => = 30% y <50%  
E = <30%

**Calificación de revisión de seguridad en código nuevo** (new\_security\_review\_rating)

Calificación de revisión de seguridad para código nuevo.

**Puntos de acceso de seguridad revisados** (security\_hotspots\_reviewed)

Porcentaje de puntos de acceso de seguridad revisados ​​(fijos o seguros).

Fórmula de proporción: Number of Reviewed (Fixed or Safe) Hotspots x 100 / (To\_Review Hotspots + Reviewed Hotspots)

**Revisión de nuevos puntos de acceso de seguridad**

Porcentaje de puntos de acceso de seguridad revisados ​​ (fijos o seguros) en código nuevo.

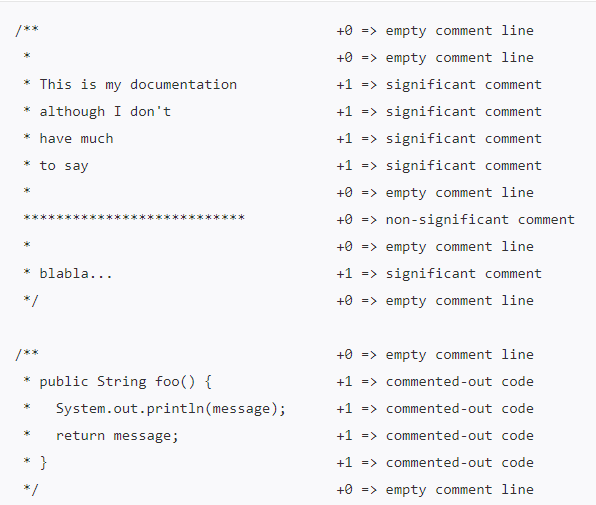
**Tamaño**

**Clases** (classes)  
Número de clases (incluidas clases anidadas, interfaces, enumeraciones y anotaciones).

**Líneas de comentario** (comment\_lines)  
Número de líneas que contienen comentarios o código comentado.

Las líneas de comentarios no significativas (líneas de comentarios vacías, líneas de comentarios que contienen solo caracteres especiales, etc.) no aumentan el número de líneas de comentarios.

El siguiente fragmento de código contiene 9 líneas de comentarios:



**Comentarios (%)** (comment\_lines\_density)  
Densidad de las líneas de comentarios = Líneas de comentarios / (Líneas de código + Líneas de comentarios) \* 100

Con tal fórmula:

* 50% significa que el número de líneas de código es igual al número de líneas de comentarios
* 100% significa que el archivo solo contiene líneas de comentarios

**Directorios** (directories)  
Número de directorios.

**Archivos** (files)  
Número de archivos.

**Líneas** (lines)  
Número de líneas físicas (número de retornos de carro).

**Líneas de código** (ncloc)  
Número de líneas físicas que contienen al menos un carácter que no es ni un espacio en blanco ni una tabulación ni parte de un comentario.

**Líneas de código por idioma** (ncloc\_language\_distribution)  
Líneas de código sin comentarios distribuidas por idioma

**Funciones** (functions)  
Número de funciones. Dependiendo del idioma, una función es una función o un método o un párrafo.

**Proyectos** (projects)  
Número de proyectos en una cartera.

**Declaraciones** (statements)  
Número de declaraciones.

**Pruebas**

**Cobertura de condición** (branch\_coverage)  
En cada línea de código que contiene algunas expresiones booleanas, la cobertura de condición simplemente responde a la siguiente pregunta: "¿Se ha evaluado cada expresión booleana como verdadera y falsa?". Esta es la densidad de posibles condiciones en las estructuras de control de flujo que se han seguido durante la ejecución de las pruebas unitarias.

Condition coverage = (CT + CF) / (2\*B) dónde:

* CT = condiciones que se han evaluado como 'verdaderas' al menos una vez
* CF = condiciones que han sido evaluadas como 'falsas' al menos una vez
* B = número total de condiciones

**Cobertura de condición en código nuevo** (new\_branch\_coverage)  
Idéntica a la cobertura de condición, pero restringida a código fuente nuevo / actualizado.

**Aciertos de cobertura de condición** (branch\_coverage\_hits\_data)  
Lista de condiciones cubiertas.

**Condiciones por línea** (conditions\_by\_line)  
Número de condiciones por línea.

**Condiciones cubiertas por línea** (covered\_conditions\_by\_line)  
Número de condiciones cubiertas por línea.

**Cobertura** (coverage)  
Es una combinación de cobertura de línea y cobertura de condición. Su objetivo es proporcionar una respuesta aún más precisa a la siguiente pregunta: ¿Cuánto del código fuente ha sido cubierto por las pruebas unitarias?

Coverage = (CT + CF + LC)/(2\*B + EL) dónde:

* CT = condiciones que se han evaluado como 'verdaderas' al menos una vez
* CF = condiciones que han sido evaluadas como 'falsas' al menos una vez
* LC = líneas cubiertas = líneas *a* cubrir - líneas\_ descubiertas
* B = número total de condiciones
* EL = número total de líneas ejecutables (lines\_to\_cover)

**Cobertura en código nuevo** (new\_coverage)  
Idéntica a la cobertura, pero restringida a código fuente nuevo / actualizado.

**Cobertura de línea** (line\_coverage)  
En una línea de código determinada, la cobertura de línea simplemente responde a la siguiente pregunta: ¿Se ha ejecutado esta línea de código durante la ejecución de las pruebas unitarias? Es la densidad de líneas cubiertas por pruebas unitarias:

Line coverage = LC / EL  
dónde

* LC = líneas cubiertas (lines\_to\_cover- uncovered\_lines)
* EL = número total de líneas ejecutables (lines\_to\_cover)

**Cobertura de línea en código nuevo** (new\_line\_coverage)  
Idéntica a la cobertura de línea, pero restringida a código fuente nuevo / actualizado.

**Aciertos de cobertura de línea** (coverage\_line\_hits\_data)  
Lista de líneas cubiertas.

**Líneas para cubrir** (lines\_to\_cover)  
Número de líneas de código que podrían cubrirse mediante pruebas unitarias (por ejemplo, las líneas en blanco o las líneas de comentarios completos no se consideran líneas para cubrir).

**Líneas para cubrir en código nuevo** (new\_lines\_to\_cover)  
Idéntico a Líneas para cubrir, pero restringido a código fuente nuevo / actualizado.

**Pruebas** unitarias omitidas (skipped\_tests)  
Número de pruebas unitarias omitidas.

**Afecciones descubiertas** (uncovered\_conditions)  
Número de afecciones que no están cubiertas por pruebas unitarias.

**Condiciones descubiertas en código nuevo** (new\_uncovered\_conditions)  
Idénticas a las condiciones descubiertas pero restringidas a código fuente nuevo / actualizado.

**Líneas descubiertas** (uncovered\_lines)  
Número de líneas de código que no están cubiertas por pruebas unitarias.

**Líneas descubiertas en código nuevo** (new\_uncovered\_lines)  
Idénticas a las líneas descubiertas pero restringidas a código fuente nuevo / actualizado.

**Pruebas unitarias** (tests)  
Número de pruebas unitarias.

**Duración de las pruebas unitarias** (test\_execution\_time)  
Tiempo necesario para ejecutar todas las pruebas unitarias.

**Errores de prueba unitaria** (test\_errors)  
Número de pruebas unitarias que han fallado.

**Fallos de pruebas unitarias** (test\_failures)  
Número de pruebas unitarias que han fallado con una excepción inesperada.

**Densidad de éxito de la prueba unitaria (%)** (test\_success\_density)  
Test success density = (Unit tests - (Unit test errors + Unit test failures)) / Unit tests \* 100