



## **TALLER 5: MÉTRICAS DE CALIDAD SW 2**

**SAMANTHA ELIANA ZAMORA AMAYA - 1151795**

**SUSANA ROJAS TRIANA – 1151767**

**ABISAID GÓMEZ CÁRDENAS - 1151810**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER**

**INGENIERÍA DE SISTEMAS**

**CÚCUTA NORTE DE SANTANDER**

**2022**



## Métricas 2 SW

### Objetivo

La medición es el proceso mediante el cual se pretenden describir entidades en el mundo real asignándoles números a sus atributos con reglas de asignación no ambiguas. La medición es clave para: tomar decisiones informadas acerca de riesgos, mejorar la calidad de los productos y predecir acertadamente recursos. Además, para lograr medidas acertadas de calidad y productividad es necesario recoger y analizar métricas de ciclos de vida previos. Las métricas ofrecen soporte cuantitativo para la toma de decisiones administrativas a lo largo del ciclo de vida del software. Sin embargo, ningún plan de recolección de métricas es factible si requiere un cómputo manual pesado. Las métricas deben recolectarse iterativa y automáticamente.

Para ello, debe: Cada equipo de desarrollo tiene un proyecto, para ello debe explicar las principales métricas de calidad de código e interpretar los resultados de estas métricas sobre sus proyectos a través de la herramienta SonarQube.

1. Dar ejemplos de uso de diversos tipos de métricas (según el proyecto en desarrollo)

La norma ISO 9126 se define como un estándar internacional, se publicó en 1992, y define los propósitos para la evaluación de la calidad de software, como la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software. Este estándar está fraccionado en cuatro partes, las cuales se encargarán de dirigir, las métricas externas, las métricas internas, la calidad en las métricas de uso y expendido.

Los modelos de calidad para el software se describen así:

- Calidad interna y externa: Pormenoriza los detalles mediante la aplicación de 6 características para calidad interna y externa. Cuando el software se utiliza como una parte de un sistema informático global, estas divisiones se aprecian externamente, y se presentan como el resultado de atributos internos de software.
- Calidad en uso: es el resultado final que contempla el cliente, después de aplicar las 6 características de la calidad interna y externa del software. También se especifican 4 características para la calidad en uso.



## 2. Describir la utilidad de las métricas de software

La métrica en software nos permite poder llevar un control de calidad en torno al ciclo de vida de este, y también nos permite poder evaluar a los profesionales que están involucrados con el proyecto. Al tener un control "real" de estos puntos podemos hacer comparativas para ver que tan bien nos está yendo o si debemos hacer mejoras en algún punto.

## 3. Explicar la importancia de la medición en los procesos de calidad

Lo primero que necesitamos de un determinado proceso o de un determinado producto, etc. que estemos desarrollando, es poderlo conocer, saber cómo es, cuáles son sus características. Todo debe definirse para poder dotarlo de características que lo hagan identificable frente a otros procesos y que nos permitan posicionarlo en un determinado orden con respecto a ellos. La definición de un determinado proceso o producto, es el primer punto que nos aporta una métrica, estableciendo una línea base.

Esta línea base aportada es la que nos permite un segundo punto también importante, la comparación entre procesos y/o productos. Aquí tenemos dos vertientes: la primera la comparación entre productos distintos o procesos distintos para determinar de una forma objetiva cuál de ellos es el mejor o el más adecuado para nuestras circunstancias y la segunda la comparación de un proceso con la evolución de sí mismo para poder determinar cuál es el grado de mejoría que se ha alcanzado a la hora de implementar una serie de mejoras y acciones evolutivas/correctivas. En resumen, la medición nos permite una comparación objetiva de productos y procesos.

Entre las principales razones para medir encontramos:

- Indicar la calidad del producto.
- Evaluar la productividad de la gente que desarrolla el producto.
- Establecer una línea de base para la estimación.
- Ayudar a justificar el uso de nuevas herramientas o de formación adicional.
- Evaluar los beneficios derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- Diferenciar las principales métricas de productos de software orientados a objetos La ingeniería es una disciplina cuantitativa, los ingenieros utilizan valores numéricos que les ayudan a modelar y obtener resultados de los productos o servicios que construyen o desarrollan.



4. Diferenciar las principales métricas de productos de software orientados a objetos

**Acoplamiento:** El acoplamiento se ve como una medida del incremento de la complejidad, reduciendo el encapsulamiento y su posible reutilización; limita, por tanto, la facilidad de comprensión y de mantenimiento del sistema.

- **Cohesión:** Número de grupos de métodos locales que no acceden a atributos comunes.
- **Valoración:** Indica la calidad de la abstracción hecha en la clase. Usa el concepto de grado de similitud de métodos. Si no hay atributos comunes, el grado de similitud es cero. Una baja cohesión incrementa la complejidad y por tanto la facilidad de cometer errores durante el proceso de desarrollo. Estas clases podrían probablemente ser divididas en dos o más subclases aumentando la cohesión de las clases resultantes. Es deseable una alta cohesión en los métodos dentro de una clase, ya que ésta no se puede dividir fomentando el encapsulamiento. Se destacan dos problemas con esta métrica:
  1. Dos clases pueden tener LCOM=0, mientras una tiene más variables comunes que la otra.
  2. No se dan guías para la interpretación de esta métrica.
  3. Nivel de granularidad: clase.
  4. Ciclo de vida: diseño, implementación.

**Complejidad:** RFC es el cardinal del conjunto de todos los métodos que se pueden invocar como respuesta a un mensaje a un objeto de la clase o como respuesta a algún método en la clase. Esto incluye a todos los métodos accesibles dentro de la jerarquía de la clase. Es decir, RFC es el número de métodos locales a una clase más el número de métodos llamados por los métodos locales.

- **Valoración:** para los autores RFC es una medida de la complejidad de una clase a través del número de métodos y de su comunicación con otras, pues incluye los métodos llamados desde fuera de la clase. Cuanto mayor es RFC, más complejidad tiene el sistema, ya que es posible invocar más métodos como respuesta a un mensaje, exigiendo mayor nivel de comprensión, lo que implica mayor tiempo y esfuerzo de prueba y depuración. Se ha señalado que la definición de esta métrica es ambigua y fuerza al usuario a interpretarla.



**Métodos ponderados por clase:** Dada una clase C1 con los métodos M1, ..., Mn y c1, ..., cn la complejidad de los métodos, WMC se define como el sumatorio de las complejidades de cada método de una clase. Si todos los métodos son considerados de igual complejidad, entonces  $c1=1$  y  $WMC=n$  (número de métodos).

- **Valoración:** Describe la complejidad algorítmica de una clase en términos de las complejidades de todos sus métodos. Está ligada a la calidad de la definición de complejidad de un método (ci). Los autores la simplifican asignando 1 a cada método, convirtiéndose así en un simple contador del número de métodos dentro de una clase. En este caso habría que considerarla como una medida del tamaño de una clase y no de complejidad, ya que una clase puede tener pocos métodos, pero muy complejos y otra clase puede tener muchos métodos, pero muy simples. Puede servir como un indicador de que una clase determinada necesite una descomposición adicional en varias clases.

**Encapsulamiento:** Es la proporción entre la suma de los grados de invisibilidad de los métodos en todas las clases y el número total de métodos definidos en el sistema. El grado de invisibilidad de un método es el porcentaje sobre el número total de clases desde las cuales un método no es visible. Es decir, MHF es la proporción entre los métodos definidos como protegidos o privados y el número total de métodos.

- **Valoración:** MHF se propone como una medida de encapsulamiento, cantidad relativa de información oculta. Cuando se incrementa MHF, la densidad de defectos y el esfuerzo necesario para corregirlos debería disminuir. No se consideran los métodos heredados.

**Herencia:** El uso de la herencia debe entenderse como un compromiso entre la facilidad de reutilización que proporciona y la facilidad de comprensión y de mantenimiento del sistema, que en general, están en relación inversa. Altos niveles de herencia indican objetos complejos, los cuales pueden ser difíciles de probar y reutilizar. Bajos niveles en la herencia pueden señalar que el código está escrito en un estilo funcional, sin aprovechar el mecanismo de herencia proporcionado por la orientación a objetos.

- **Valoración:** es un indicador del nivel de reutilización. También se propone como ayuda para evaluar la cantidad de recursos necesarios a la hora de probar.



- Nivel de granularidad: clase, programa, sistema. Ciclo de vida: diseño, implementación.
- Relacionado con: reutilización.

**Polimorfismo:** Es la proporción entre el número real de posibles situaciones polimorfas para una clase Ci y el máximo número posible de situaciones polimorfas en Ci. Es decir, es el número de métodos heredados redefinidos dividido entre el máximo número de situaciones polimorfas distintas posibles.

- Valoración: POF es una medida del polimorfismo y una medida indirecta de la asociación dinámica en un sistema. El polimorfismo se debe a la herencia. Nivel de granularidad: clase.
- Ciclo de vida: diseño, implementación.
- Relacionado con: herencia.

**Reutilización:** Se refiere al comportamiento y a las técnicas que garantizan que una parte o la totalidad de un programa informático existente se puedan emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia.

## 5. Formular metas y planes de calidad

<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Fase No. 3 Implementación - 2	Modelo de diseño, análisis de requerimientos
Actividades independientes de las fases	<ul style="list-style-type: none"><li>• Actualización del manual usuario</li><li>• Aprobación del documento de control de cambios</li></ul>
Actividades iniciales de iteración	<ul style="list-style-type: none"><li>• Cronograma de actividades</li><li>• Asignación de actividades</li><li>• Control de cambios</li></ul>
Actividades iniciales del proyecto	<ul style="list-style-type: none"><li>• Cronograma de iteración</li><li>• Actualización de documento técnico</li></ul>



Código de tarea	Código: F3. Implementación
Actividad	Elaboración y aprobación de las actividades de la iteración a implementar para el seguimiento de las métricas de calidad establecidas en el proyecto

6. Explicar la importancia de los datos históricos en los modelos predictivos.

El Análisis Predictivo es una de las herramientas que forman parte de un conjunto de técnicas más amplio conocido como Business Intelligence, el uso del Análisis Predictivo es consecuencia de la nueva cultura que se ha generalizado con respecto a los datos. La capacidad real de almacenar y procesar grandes conjuntos de datos, ligada a los avances experimentados por las TI, ha permitido generar archivos masivos de datos de todo tipo, susceptibles de ser analizados en busca de tendencias.

El análisis predictivo se fundamenta en la identificación de relaciones entre variables en eventos pasados, para luego explotar dichas relaciones y predecir posibles resultados en futuras situaciones. Ahora bien, hay que tener en cuenta que la precisión de los resultados obtenidos depende mucho de cómo se ha realizado el análisis de los datos, así como de la calidad de las suposiciones. “Toda esta información abre nuevas perspectivas y permite habilitar servicios innovadores basados en las tecnologías emergentes como el tratamiento de grandes volúmenes de información, la minería de datos, el análisis predictivo, etc.”

Los modelos predictivos que se centran en predecir el comportamiento de un cliente en particular, los modelos descriptivos identifican diferentes relaciones entre los clientes y los productos. Los datos son la fuente de la que se obtienen las variables, las relaciones entre ellas, el conocimiento inducido o los patrones de comportamiento identificados, convirtiéndose en un elemento vital de todo análisis predictivo.



7. Aplicar las métricas de software para asegurar la calidad del producto de manera objetiva.

Son métricas como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

1. Métricas de mantenibilidad
2. Métricas de testeabilidad
3. Métricas de flexibilidad
4. Métricas de corrección
5. Métricas de defectos
  - Medición de remoción de defectos de software
  - Medición de la eficacia de la remoción de defectos
  - Defectos informados por los clientes
  - Medición de la eficacia de la eliminación de defectos
6. Métricas de facilidad de uso
7. Métricas de integridad
8. Métricas de facilidad de mantenimiento
  - Índice de mantenibilidad

8. Identificar mecanismos de evaluación y adecuación de un proyecto.

***Identificar el objeto y alcance del sistema de seguimiento y evaluación.***

El propósito y el alcance del sistema de seguimiento y evaluación responden a dos preguntas: ¿por qué necesitamos un sistema de seguimiento y evaluación? y ¿qué alcance debe tener? Esto sirve como punto de referencia del sistema de seguimiento y evaluación, ya que guía las principales decisiones, como las necesidades de información, los enfoques metodológicos, el fortalecimiento de la capacidad y la asignación de recursos. A continuación, se bosqueja algunos aspectos importantes que deben tenerse en cuenta al determinar el objeto y el alcance de este sistema.

***Planificar la recopilación y gestión de datos.***

Una vez definidas las necesidades de información del proyecto o programa, el siguiente paso consiste en planificar la recopilación y la gestión confiables de los





datos a fin de que puedan analizarse eficientemente y utilizarse como información. Tanto la recopilación como la gestión de los datos están íntimamente ligadas, ya que la gestión comienza cuando se recopila el primer dato.

### ***Planificar el análisis de datos.***

El análisis de los datos es el proceso mediante el cual los datos recopilados (en bruto) se convierten en información utilizable. Constituye una etapa crucial del proceso de planificación del seguimiento y la evaluación, porque da forma a la información comunicada y determina su posible utilización. En realidad, constituye un proceso continuo durante todo el ciclo del proyecto o programa en el que los datos recopilados cobran sentido y permiten fundamentar los programas en marcha y futuros. Este análisis puede llevarse a cabo cuando los datos se recopilan inicialmente y, por supuesto, cuando se da cuenta de ellos en el proceso de elaboración de informes (analizado en la siguiente etapa). El análisis de los datos supone buscar tendencias, grupos temáticos u otras relaciones entre diferentes tipos de datos, evaluar el desempeño en función de los planes y las cotas, extraer conclusiones, prever los problemas, e identificar las soluciones y las prácticas óptimas para la adopción de decisiones y el aprendizaje institucional. El análisis confiable y oportuno es esencial para la credibilidad y la utilización de los datos

### **Planificar cómo se difundirá y utilizará la información.**

Después de haber definido las necesidades de información del proyecto o programa y la forma en que se recopilan, gestionarán y analizarán los datos, el siguiente paso consiste en planificar cómo volcar los datos en un informe y darles buen uso. La elaboración de informes es la parte más visible del sistema de seguimiento y evaluación, pues aquí es donde los datos recopilados y analizados se presentan como información para ser utilizada por los principales interesados. La elaboración de los informes constituye un componente crucial del seguimiento y la evaluación porque, independientemente de cuán bien se hayan recopilado y analizado los datos, si su presentación es precaria, el informe no será de mucha utilidad – lo que se traduce en un derroche considerable de recursos, tiempo y personal de gran valor –. Lamentablemente, hay muchos ejemplos de datos valiosos que terminan siendo inservibles debido a la elaboración de informes deficientes.



## ***Planificar el fortalecimiento de la capacidad y los recursos humanos necesarios para las actividades de seguimiento y evaluación.***

Un sistema de seguimiento y evaluación eficaz requiere gente idónea que lo sustente. Si bien el plan de seguimiento y evaluación determina las responsabilidades inherentes a la recopilación de datos sobre cada indicador, también es importante designar a los responsables de los procesos de seguimiento y evaluación, incluida la gestión y el análisis de datos, la elaboración de informes y la capacitación en seguimiento y evaluación. Esta sección resume los principales aspectos que deben tenerse en cuenta a la hora de planificar el fortalecimiento de la capacidad y los recursos humanos del sistema de seguimiento y evaluación de un proyecto o programa

## ***Preparar el presupuesto de seguimiento y evaluación.***

Lo mejor es comenzar a planificar el presupuesto de seguimiento y evaluación en forma sistemática durante la primera etapa del proceso de formulación del proyecto o programa, a fin de asignar los fondos suficientes a estas actividades. La siguiente sección resume los principales aspectos que deben tenerse en cuenta al elaborar el presupuesto.