# MÉTRICAS DE PROCESO Y PROYECTO

*Enma E. Salazar*[1], *Manuel F. Salazar*[2]
Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación

***Resumen*** *— El proceso de software y las métricas del proyecto son medidas cuantitativas que proporcionan a los ingenieros de software una amplia visión del proceso y un conocimiento detallado acerca del proyecto que se lleva a cabo utilizando el proceso como marco de trabajo. La medición permite destacar las tendencias (ya sean buenas o malas) y hacer mejores estimaciones que conducirán a un proyecto exitoso; comienza definiendo un conjunto limitado de medidas del proceso y del proyecto, las cuales por lo general se normalizan empleando métricas orientadas al tamaño o la función, el resultado se analiza y compara con promedios pasados, luego se valoran las tendencias y se generan conclusiones.*

***Palabras Claves*** *— Indicador, Medición, Medida, Métrica, Punto de función, Error, Defecto*

## INTRODUCCIÓN

Medir el *software* no es una actividad sencilla. Se puede decir que toda magnitud física se puede medir, pero el software es, sobre todo, el resultado de una actividad básicamente intelectual, tanto en el origen como en el resultado final.

En el mundo de la ingeniería del SW encontramos dificultades en ponernos de acuerdo sobre que medir y como evaluar las medidas. Por tanto, se debe disponer, en primer lugar, de diferentes unidades de medida que puedan ser útiles para caracterizar y medir varias funcionalidades del software que se quiere implentar: el tamaño del proyecto, la evaluación de la calidad y la fiabilidad, etc.

En este artículo se analizará algunas de las muchas métricas del *software*, con atención especial a las métricas de tamaño (LOC) y funcionalidades (puntos de función), que son las más ampliamente utilizadas.

## MÉTRICAS DE PROCESO Y PROYECTO

Hay cuatro razones para medir: Caracterizar, Evaluar, Predecir y Mejorar.

Aunque medida, medición y métrica son términos que suelen usarse de manera intercambiable, es importante observar sus diferencias.

- **Medida:** Valor asignado a un atributo de una entidad mediante una medición.
  Ejemplo: 35.000 líneas de código
- **Medición:** Es el acto de determinar una medida.
  Ejemplo: Ana será la encargada de medir las LDC de cada módulo del sistema.
- **Métrica:** Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición.
  Ejemplo: La productividad de este proyecto fue de 500 líneas (LDC/persona-mes)
- **Indicador:** Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software.
  Ejemplo: La productividad media de nuestra empresa es de 500 (LDC/pm)

Las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo y el producto para intentar aumentar su calidad.

## MÉTRICAS EN LOS DOMINIOS DEL PROCESO Y EL PROYECTO

Las métricas del proceso se recopilan en el curso de todos los proyectos y durante largos períodos, permiten:

Al gestor:

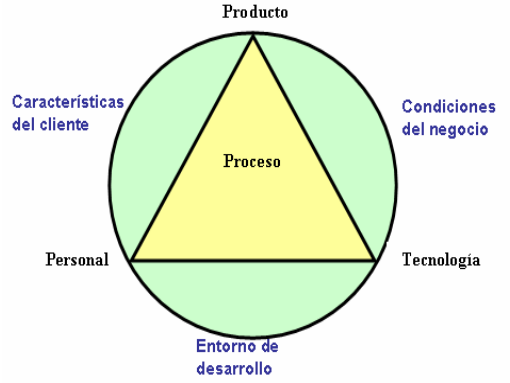
- Evaluar lo que funciona y lo que no.
- Obtener un conjunto de indicadores de proceso que conduzcan a la mejora de los procesos en sí, esto es a largo plazo.

A la organización

- Adoptar una visión estratégica.

El proceso solo es uno de varios factores controlables en la mejora de la calidad del software y el desempeño organizacional.

FIGURA 1
EL PROCESO Y LOS DIVERSOS FACTORES DE UN PROYECTO




[1] Enma E. Salazar, UTPL, Loja, eesalazar@utpl.edu.ec
[2] Manuel F. Salazar, UTPL, Loja, mfsalazarx@utpl.edu.ec

¿Por qué nos centramos en mejorar el proceso? Porque el proceso es un factor clave y controlable para mejorar la calidad del software y el rendimiento de la organización.

En la Figura 1 el proceso está en el centro de un triángulo que conecta tres factores influyentes en la calidad del software: Personal, Producto y Tecnología, él triángulo a su vez se encuentra dentro de un círculo de condiciones ambientales como: entorno de desarrollo, condiciones del negocio y características del cliente.

Al medir el proceso, existen usos privados y públicos para las métricas de software.

- Métricas privadas (de uso individual)
  - Índices de defectos (individual, por módulo)
  - Errores encontrados durante el desarrollo
- Métricas Públicas (para el equipo)
  - Índices de defectos
  - Errores encontrados en revisiones técnicas del proyecto
  - LDC
  - Puntos de función por módulo y función

Las métricas del proyecto permiten:

Al gestor:

- Valorar el estado de un proyecto en curso
- Rastrear los riesgos potenciales
- Detectar áreas problemáticas antes de que se conviertan en críticas.
- Ajustar el flujo y las tareas de trabajo.
- Evaluar la habilidad del equipo en controlar la calidad de los productos de trabajo de la IS.

A la organización:

- Adoptar una visión táctica.

Las métricas del proyecto se emplean básicamente con dos fines: minimizar el tiempo de desarrollo y valorar la calidad del producto sobre una base actual.

## MEDICIÓN DEL SOFTWARE

La medición del software se clasifica en dos categorías:

1) **Métricas Directas:** Aquellas que no depende de ninguna métrica de otro atributo.
   Ejemplo: costo, esfuerzo, líneas de código, velocidad de ejecución, número de defectos
2) **Métricas Indirectas:** Son aquellas que se obtienen a partir de métricas directas.
   Ejemplo: funcionalidad, calidad, complejidad, eficiencia, fiabilidad, etc.

### Métricas orientadas al tamaño

Consideran el tamaño de software que se ha producido, siendo por ello las líneas de código (LDC) el valor de normalización. Son medidas directas del resultado y del proceso.

Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño como la siguiente:

TABLA 1
TABLA DE VALORES ORIENTADOS AL TAMAÑO

| Proyecto | Esfuerzo | Miles $ | KLDC | Págs. Doc | Errores | Defectos |
|---|---|---|---|---|---|---|
| A | 24 | 168 | 12,1 | 365 | 29 | 3 |
| B | 62 | 440 | 27,2 | 1124 | 86 | 5 |
| C | 43 | 314 | 20,2 | 1050 | 64 | 6 |

Medidas

- Líneas de código (LDC).
- Esfuerzo en persona-mes.
- Costo en dólares.
- Número de páginas de documentación.
- Número de errores. Fallas detectadas antes de entregar el software al cliente.
- Número de defectos. Fallas detectadas después de entregar el software al cliente.
- Número de personas en el proyecto.” [2]

Métricas

- Productividad = KLDC/Esfuerzo
- Calidad = errores/KLDC (miles de líneas de código)
- Costo = $/KLDC
- Documentación = Pags. Doc/KLDC

“Ventajas

- Son fáciles de calcular.
- Muchos modelos de estimación de software usan LDC o KLDC como datos de entrada.
- Existe un amplio conjunto de datos y literatura basados en LDC.

Desventajas

- Son dependientes del lenguaje de programación.
- Perjudica a los programas cortos pero bien diseñados.
- Su uso en estimación es difícil porque hay que estimar las LDC a producirse mucho antes de que se complete el análisis y el diseño.”[2]

### Métricas orientadas a la función

Son medidas indirectas del software y del proceso. Se centran en la funcionalidad o utilidad del programa. “Emplean como un valor de normalización una medida de la funcionalidad que entrega la aplicación. La métrica orientada a la función utilizada con mayor amplitud es el punto de función (PF).”[1]

Procedimiento para calcular el Punto de Función:

1. Llenar la columna de CUENTA de la siguiente tabla de valores, donde se determinan 5 características del ámbito de la información.

TABLA 2
TABLA DE VALORES DEL DOMINIO DE LA INFORMACIÓN

| Parámetro | CUENTA | Factor de ponderación | | | Subtotal |
|---|---|---|---|---|---|
| | | Simple | Medio | Complejo | |
| Entradas de usuario | | 3 | 4 | 6 | |
| Salidas de usuario | | 4 | 5 | 7 | |
| Peticiones de usuario | | 3 | 4 | 6 | |
| Archivos | | 7 | 10 | 15 | |
| Interfaces externas | | 5 | 7 | 10 | |
| | | | | **CUENTA TOTAL** | |

- **Entradas de Usuario (Entradas):** cualquier entrada (pantalla, formulario, cuadro de diálogo, control o mensaje) a través de la cual el usuario u otro programa puede añadir, borrar o cambiar datos.
- **Salidas de Usuario (Salidas):** cualquier salida (pantalla, informe, gráfico, mensaje) que tenga un formato diferente o requiera un procesamiento diferente a otros tipos de salida, generada para el usuario u otro programa.
- **Peticiones de Usuario (Consultas):** combinaciones de entrada/salida en las que cada entrada genera una salida simple e inmediata.
- **Archivos Lógicos Internos (Archivos):** principales grupos lógicos de datos de usuarios o de control que están controlados por el programa (una tabla de un SGBDR).
- **Archivos de Interfaz Externos (Interfaces):** cada uno de los grupos de datos lógicos o información de control que entra o sale del programa.

2. Multiplicar el valor de CUENTA por el Factor de Ponderación indicado, dependiendo de su complejidad para obtener Subtotal.

Se asocia un valor de complejidad a cada medida, tomando en cuenta la heurística de la siguiente tabla.

TABLA 3
TABLA DE COMPLEJIDADES

| Tipos de archivos referenciados | Tipos de datos elementales | | |
|---|---|---|---|
| | 1 – 5 | 6 – 19 | 20+ |
| 0 – 1 | Bajo | Bajo | Medio |
| 2 – 3 | Bajo | Medio | Alto |
| 4+ | Medio | Alto | Alto |

3. Obtener CUENTA TOTAL
4. Evaluar los siguientes 14 atributos que impactan en el desarrollo, en escala de 0 a 5.

TABLA 4
FACTORES DE INFLUENCIA EN LA DIFICULTAD DEL SISTEMA

| Factor | Ejemplo |
|---|---|
| 1. Comunicaciones de datos | Una aplicación para el sector bancario, donde se requieren numerosas transacciones monetarias. |
| 2. Procesamiento distribuido | Un motor de búsqueda en Internet, donde el procesamiento está distribuido en decenas de máquinas. |
| 3. Objetivos de rendimiento | Una aplicación para el control del tráfico aéreo, que debe proporcionar continuamente información precisa sobre la posición y rumbo de los aviones. |
| 4. Configuración de uso intensivo | Un sistema para matrículas en una universidad, donde concurren cientos de alumnos al mismo tiempo. |
| 5. Tasas de transacción rápidas | Una aplicación para el sector bancario, donde deben realizarse millones de transacciones durante la noche. |
| 6. Entrada de datos en línea | Un programa en el que los datos de entrada provienen de papeles o formularios impresos. |
| 7. Amigabilidad en el diseño | Un programa de análisis financiero utilizado por el directivo de una empresa, capaz de orientarle y asesorarle. |
| 8. Actualización de datos en línea | Una aplicación para reserva de billetes, en la que deben bloquearse y modificarse ciertos registros en las BB.DD. para evitar que un mismo asiento sea vendido dos veces. |
| 9. Procesamiento complejo | Un sistema para diagnóstico médico, el cual realiza costosas operaciones de decisión lógica hasta obtener un resultado. |
| 10. Reusabilidad | Un procesador de textos en el que, por ejemplo, su barra de menús puede utilizarse desde una hoja de cálculo, un generador de informes de una base de datos, etc... |
| 11. Facilidad de instalación | Cualquier aplicación de propósito general, de tal forma que cualquier persona pueda realizar la instalación fácilmente. |
| 12. Facilidad operacional | Una aplicación para tratamiento de grandes cantidades de información, donde es muy importante la efectividad de los procesos de backup y recuperación de datos. |
| 13. Adaptabilidad | Una aplicación software para una multinacional con oficinas en varios países. |
| 14. Versatilidad | Un sistema que admite diversas situaciones de uso, tanto para facilitar los cambios como para ser utilizada por el usuario |

A cada atributo se le asignará un valor dependiendo del grado de influencia de éstos.

**Sin Influencia (0)**. El sistema no contempla este atributo.
**Influencia Mínima (1)**. La influencia de este atributo es muy poco significativa.
**Influencia Moderada (2)**. El sistema contempla este atributo y su influencia, aunque pequeña, ha de ser considerada.
**Influencia Apreciable (3)**. La importancia de este atributo debe ser tenida en cuenta, aunque no es fundamental.
**Influencia Significativa (4)**. Este atributo tiene una gran importancia para el Sistema.

**Influencia Muy Fuerte (5)**. Este atributo es esencial para el sistema y se lo debe tomar en cuenta a la hora del diseño.

5. Sumar los puntos asignados a cada factor y obtener un TOTAL GI que indica un valor de ajuste de complejidad.
6. Calcular Punto de Función, utilizando la siguiente fórmula:

*PUNTO FUNCIÓN = CUENTA_TOTAL * (0,65+ 0,01*TOTAL GI)*

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

Una vez calculado el punto de función se usan de forma analógica a las LDC como medida de la productividad, calidad y otros productos del software.

*Productividad = PF / Eficiencia*
*Calidad = Errores / PF*
*Costo = Dólares / PF*
*Documentación = Pags. Doc / PF*

### Reconciliación de las métricas LDC y PF

La relación entre líneas de código y puntos de función depende del lenguaje de programación en que se implementan el SW y la calidad del diseño.

### Métricas orientadas a objetos

No proporcionan suficiente granularidad para la planificación y los ajustes de esfuerzo. Las siguientes son métricas sugeridas para proyectos OO:

- Número de guiones de escenario
- Número de clases clave
- Número de clases de apoyo
- Número promedio de clases de apoyo por clase clave.
- Número de subsistemas.

### Métricas orientadas a casos de uso

El caso de uso se define en etapas tempranas del proceso de software, lo que permite emplearlo en la estimación antes de iniciar las actividades significativas de modelado y construcción.

### Métricas de proyectos de ingeniería Web

"El objetivo de los proyectos de ingeniería Web es construir una aplicación Web que proporcione una combinación de contenido y funcionalidad al usuario final." [1] Entre las medidas que se recopilan existen las siguientes:

- Número de páginas web estáticas
- Número de páginas web dinámicas
- Número de vínculos internos de la página
- Número de objetos de datos persistentes
- Número de sistemas externos en interfaz
- Número de objetos de contenido estático
- Número de objetos de contenido dinámico
- Número de funciones ejecutables

## MÉTRICAS PARA CALIDAD DEL SOFTWARE

Las métricas de calidad de software se enfocan sobre el proceso, el proyecto y el producto. Estas métricas las podemos dividir en dos grupos; el primer grupo se le recolecta antes de la entrega del producto y las otras luego de haberlo entregado.

### Medidas de la calidad

- **Corrección:** Es el grado en que el software desempeña la función para la que fue creado. Su medida es **nº de defectos por KLDC.**
- **Facilidad de Mantenimiento:** es la facilidad para corregir un error, adaptar un programa a cambios, o mejorarlo si el cliente desea un cambio. Su medida es TMC (tiempo medio de cambio).
- **Integridad:** Es la capacidad para resistir ataques, provocados o no, contra su seguridad, ya sea sobre programas, datos y documentos. Se la puede definir como:

$$Integridad = (amenaza\ x\ (1 - seguridad))$$

*Amenaza:* es la probabilidad de que un cierto tipo de ataque ocurra en un tiempo dado,
*Seguridad:* es la probabilidad de que se pueda contrarrestar un cierto tipo de ataque.

- **Facilidad de uso:** Se refiere a Habilidad intelectual y/o física requerida para aprender a utilizar el sistema; es decir, "amistad con el usuario".

### Eficacia en la eliminación de defectos

Habilidad de filtrar las actividades de la garantía de cualidad; cuando se considera como un proyecto se la define como:

$$EED = E / (E + D)$$

Donde, $E$ = Error y $D$ = Defecto; el valor ideal de la *EED* es 1.

La EED también se puede aplicar al proceso para valorar la habilidad de un equipo de encontrar errores antes de que pase a la siguiente actividad del marco de trabajo. En este contexto se la define como:

$$EED_i = E_i / (E_i + E_{(i+1)})$$

Donde *i* representa una actividad e *i+1* representa la siguiente actividad luego de *i*;

## INTEGRACIÓN DE LAS MÉTRICAS DENTRO DEL PROCESO DE SOFTWARE

Generalmente la recopilación de medidas en una organización genera resistencia debido a la cultura de los programadores que habitualmente evitan el "papeleo".

Es importante medir el proceso de ingeniería de software y el producto (software) que elabora ya que "lo que no se puede medir no se puede controlar", siendo de gran utilidad la Integración de métricas.

### Establecimiento de una línea base

Una *línea base de métricas* consiste de datos recopilados en proyectos previos de desarrollo de software. Con el establecimiento de una línea base se obtienen beneficios en el proceso, el proyecto y el producto.

### Recopilación, cálculo y evaluación de métricas

El proceso con el que se establece una línea base de métricas lo podemos observar en la presente figura.

FIGURA 2
PROCESO DE RECOPILACIÓN DE MÉTRICAS DE SOFTWARE


Primeramente recopilamos los datos de proyectos previos, así obtenemos las medidas, las cuales utilizamos para calcular las métricas. Dichas métricas deben evaluarse y aplicarse durante la estimación del trabajo técnico, produdiendo así un conjunto de indicadores que guían el proceso o proyecto.

## MÉTRICAS PARA ORGANIZACIONES PEQUEÑAS

Una organización pequeña puede seleccionar el siguiente conjunto de medidas que se recopilan con facilidad:

- $t_{cola}$: Tiempo desde solicitud hasta que evaluación está completa
- $T_{eval}$: Esfuerzo para realizar evaluación (persona-horas)
- $t_{eval}$: Tiempo desde que se completa la evaluación hasta la asignación de pedido de cambio
- $T_{cambio}$: Esfuerzo para hacer el cambio (persona-horas)
- $t_{cambio}$: Tiempo requerido para hacer el cambio (persona-horas)
- $E_{cambio}$: Errores descubiertos durante el trabajo para hacer el cambio.
- $D_{cambio}$: Defectos descubiertos después de que el cambio es liberado.

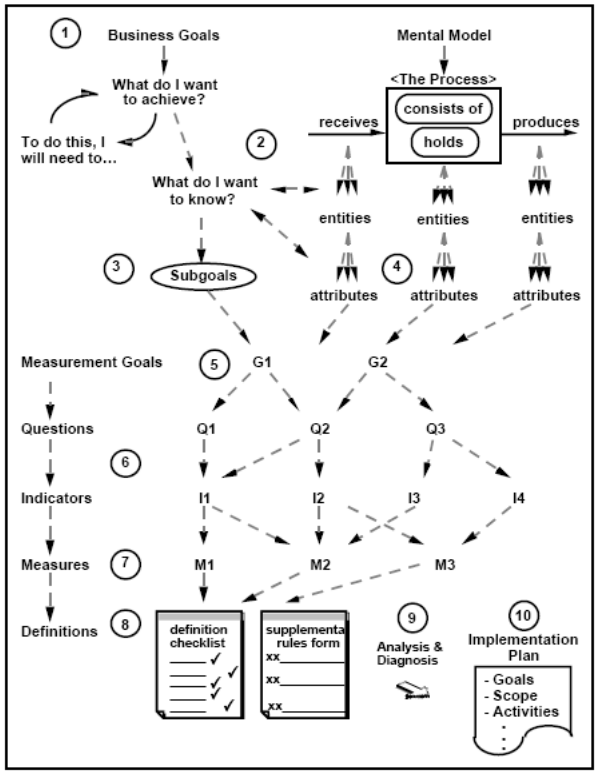
La Eficacia en la Eliminación de Defectos (EDD) se calcula como:

$$EED = E_{cambio}/(E_{cambio}+D_{cambio})$$

## ESTABLECIMIENTO DE UN PROGRAMA DE MÉTRICAS DE SOFTWARE

Existen varios enfoques para establecer un programa de métricas dentro de una organización de software.

FIGURA 3
MODELO DE UN PROCESO PARA LA SELECCIÓN DE MEDIDAS DE SOFTWARE, SEGÚN EL SEI [6]




La Software Engineering Institute (SEI) ha elaborado una guía detallada "dirigida por metas", que sugiere los siguientes pasos:

1. "Identificar los objetivos de la empresa.
2. Identificar lo que se quiere conocer o aprender.
3. Identificar los subobjetivos.
4. Identificar las entidades y a tributos relacionados con los objetivos secundarios.

5. Formalizar los objetivos de la medición.
6. Identificar preguntas cuantificables y los indicadores relacionados que se emplearán como apoyo para lograr los objetivos de sus mediciones.
7. Identificar los elementos de datos que se recopilarán para construir los indicadores que ayudarán a responder las preguntas.
8. Definir las medidas que se emplearán y hacer que estas definiciones sean operativas.
9. Identificar las acciones que se tomarán para implementar las medidas.
10. Preparar un plan para implementar las medidas." [6]

## RESULTADOS

- Clara diferenciación entre métricas del proceso y del proyecto.
- Concienciación de la gran importancia que tiene el uso de métricas de software.

## CONCLUSIONES

- Medir el software no sólo es útil, sino necesario.
- La métrica NO puede interpretar por sí sola un concepto medible, por ello existe la necesidad de indicadores.
- Medición "objetiva antes que subjetiva".
- Mejoramos el proceso porque es controlable.
- Las Métricas del Proceso se consideran estratégicas y las Métricas del proyecto tácticas.
- Las métricas de calidad de software se enfocan sobre el proceso, el proyecto y el producto.
- Con una línea base de métricas los ingenieros de software y los gestores pueden comprender mejor el trabajo y el producto que elaboran.

## BIBLIOGRAFÍA


[1] PRESSMAN, Roger, "Ingeniería del Software. Un enfoque práctico". Sexta edición. McGrawHill Interamericana Mexico, 2005.

[2] CABRERA, Armando, "Ingeniería del Software". Primera Edición, Universidad Técnica Particular de Loja. Loja – Ecuador, 2006.

[3] Métricas del Proyecto de Software, (en línea), disponible en http://www.vision.uji.es/~sanchez/Teach/PDF-E77/Tema1.pdf

[4] OTONIEL, Giraldo, "Métricas, Estimación y Planificación en Proyectos de Software", (en línea), disponible en http://www.willydev.net/Descargas/WillyDEV_PlaneaSoftware.Pdf

[5] VARAS, Marcela, "Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo", Universidad de Concepción, (en línea), disponible en http://www.inf.udec.cl/~mvaras/papers/arica/arica.htm

[6] Park, R. E., W. B. Goether y W. A. Florac, Goal Driven, "Software Measurement – A Guide Book", CMU/SEI-96-BH-002, Software Engineering Institute, Carnegie Mellon University, agoso de 1996.