

Instituto Tecnológico de las Américas  
(ITLA)



## *“Proceso de Software y Métricas del Proyecto”*

Elaborado Por:

Brian Gómez (2016-3557)  
Christopher Santana (2016-3560)  
Máximo Bautista (2016-3538)  
Saul de Jesús Castro Barrera (2016-3561)  
Jeremy Arias González. (2016-3572)

Período académico:

C2

Fecha de Entrega:

Sábado 11/08/2018

PROFESOR  
Leandro Fondeur

Santo Domingo, República Dominicana D.N.

Agosto 2018

# **1. Describa con sus palabras la diferencia entre métricas de proceso y de proyecto.**

Además de la diferencia básica de que las de proceso son estratégicas y las de proyecto tácticas, también tiene como diferencias que las de proceso se utilizan para prevenir errores o fallas en el desarrollo del proyecto antes de que se dé inicio a su desarrollo por lo que nos ayuda a tener un desarrollo más hábil mientras que las de proyecto es ya cuando estamos trabajando y estas nos ayuda a corregir defectos y de más faltas que se vayan encontrado durante el desarrollo del proyecto.

Se debe tomar en cuenta que las métricas del proyecto provienen de las antes establecidas métricas de proceso.

Se puede agregar que las **métricas del proceso** determinan el curso del proceso de producción de software y las del **proyecto** determinan el curso del proyecto actual.

### **3. ¿Qué es una medida indirecta y por qué tales mediciones son comunes en el trabajo con métricas de software?**

La medida indirecta en el software se refiere a aquellas medidas que incluyen la funcionalidad, calidad, complejidad, eficiencia, confiabilidad, capacidad de mantenimiento, y muchas otras más, las cuales son muy difíciles de valorar y solo se pueden estimar de manera indirecta.

Las mediciones de este tipo son comunes en el trabajo con métricas de software, porque son necesarias para dar un enfoque sobre la calidad que tiene el software.

**6. Presente un argumento contra las líneas de código como medida para la productividad del software. ¿Su caso se sostendría cuando se consideren decenas o cientos de proyectos?**

Se pueden usar las líneas de código para evaluar la productividad, sin embargo, no siempre han de arrojar el resultado más confiable. Una de las cosas a considerar es que las líneas de código dependen de las tecnologías a utilizar, ya que una evaluación que considerara que una mayor cantidad de líneas de código implica un aumento en la productividad en la realización del software, podría equivocarse si al evaluar fragmentos del código más importantes a nivel de productividad, ya que resuelven un problema importante de una manera elegante tienen menos líneas de código que un fragmento de código repetitivo quizás necesario como base para la implementación de una estructura requerida y repetitiva dentro del proyecto. Esto arrojaría un resultado erróneo en la estimación de productividad del software, por hacer uso de la cantidad de líneas de código como indicativo de la productividad del software.

**8. El software que se usa para controlar una fotocopidora requiere 32 000 líneas de C y 4 200 líneas de Smalltalk. Estime el número de puntos de función para el software dentro de la fotocopidora.**

En C, un LOC= 162 puntos de función en promedio.

Si tenemos 32,000 LOC entonces:

**$32,000 * 162 = 5,184,000$  PF en C.**

En Smalltalk, un LOC = 26 PF en promedio.

Si tenemos 12,000 LOC entonces:

**$12,000 * 26 = 312,000$  PF.**

Entonces tenemos un total promedio de:

$5,184,000 + 312,000 = 5,496,000 = \underline{\underline{2,748,000 \text{ PF en promedio}}}$

**11. En la conclusión de un proyecto, se determinó que se encontraron 30 errores durante la actividad de modelado y 12 durante la actividad de construcción, que fueron rastreables en errores que no se descubrieron en la actividad de modelado. ¿Cuál es la ERD para la actividad de modelado?**

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

Dónde: E = 30 y D = 12

Formula de ERD = 30/ (30+12)

$$ERD = 30/42$$

$$ERD = 0.714$$