



TÍTULO DEL PROYECTO:

Cuestionario sobre Métricas de Software

ESTUDIANTE:

Colque García Ariel Rodrigo

CARRERA: Ing. Ciencias De la Computación

DOCENTE: Ing. Angel Hilmar Baspineiro Valverde

MATERIA: SIS (325)

Sucre-Bolivia

1. **¿Porque la métrica basada en Líneas de Código (LDC) para el tamaño no sería una buena métrica en el contexto actual de los lenguajes de desarrollo? Mencione al menos 3 razones.**

La medición del tamaño se basa en líneas de cálculo de código basado en la medida directa de un Software. Pero algunas desventajas que esto presenta en el contexto actual de los lenguajes de desarrollo son:

- Esta medida depende del lenguaje de programación.
- A veces, es muy difícil estimar la cantidad de Líneas de Código en una etapa temprana de desarrollo.
- No puede medir el tamaño de la especificación tal como se define en el código.
- No puede medir el tamaño de la especificación tal como se define en el código.
- Aunque es simple de medir, es muy difícil de entender para los usuarios.

2. **¿Cuál es la ventaja de la métrica basada en Puntos Función frente a las basadas en LDC, para medir el tamaño de la aplicación**

Como se mencionó que las métricas basadas en LDC tienen algunas desventajas, los puntos de función nos ayudan a medir el tamaño del software de forma rápida, confiable, realizando estimaciones de esfuerzo exactas basadas en un modelo paramétrico y no subjetividades.

3. **Según el video que proceso y que cálculos se siguieron para obtener los puntos función de la aplicación de ejemplo.**

Para el cálculo de los puntos de Función se utilizó la siguiente fórmula:

$$PF = \text{cuenta-total} \times [0.65 + 0.01 \times \Sigma(Fi)]$$

Posteriormente se procede a obtener los diferentes valores de dominio:



Para que al final se haga el cálculo correspondiente:

Valor dominio de información	Conteo		Factor ponderado			
			Simple	Promedio	Complejo	
Entradas externas (EE)	3	×	3	4	6	= 9
Salidas externas (SE)	2	×	4	5	7	= 8
Consultas externas (CE)	2	×	3	4	6	= 6
Archivos lógicos internos (ALI)	1	×	7	10	15	= 7
Archivos de interfaz externos (AIE)	4	×	5	7	10	= 20
Conteo total		→	50			

$$PF = \text{cuenta-total} \times [0.65 + 0.01 \times \Sigma(Fi)]$$

Si asumimos $\Sigma(Fi) = 46$, entonces:

$$PF = 50 \times (0.65 + 0.01 \times 46) = 55,5$$

4. Mencione algunos ejemplos de métricas directas, ósea ejemplos de mediciones de atributos que se pueden medir directamente sin necesidad de medir otros atributos. Y mencione si hay alguna ecuación (o regla) que se utiliza.

Algunos ejemplos de Métricas directas son;

- Complejidad ciclomática
- Líneas de código
- Cantidad de errores, de defectos
- número de defectos
- duración del proceso de prueba

En a las gunas métricas directas si se usa un formula directa por ejemplo para la medición de tiempo de respuesta su fórmula es:

$X = \text{tiempo (calculado o simulado)}$

5. ¿Por qué la medición del tamaño por puntos función sería una métrica indirecta?

Porque se requiere otros atributos para tener el valor esperado.

6. Las métricas pueden clasificarse también en métricas del proceso, proyecto y producto. Mencione un ejemplo de cada categoría, mostrando el atributo a medir, la regla (o fórmula, modelo, criterio) y el valor

En la **métrica de proceso** un ejemplo serio la de integridad del sistema que se puede definir como: $\text{Integridad} = C [(I - \text{amenaza}) \times (1 - \text{seguridad})]$ Donde se suman la amenaza y la seguridad para cada tipo de ataque.

Métrica de Proyecto: Velocidad se calcula por: Número de puntos ó valor de historias completados en la iteración.

Métricas de Producto: Longitud del código Ésta es una medida del tamaño del programa.

7. Si tendríamos que medir la confiabilidad del software, que otros atributos de bajo nivel deberíamos medir primero según el modelo ISO9126

Tendríamos que medir primero el Nivel de Madurez, Tolerancia a fallas y de Recuperación.

8. Buscar una métrica para el software orientada al tamaño y que se use en procesos ágiles. Una métrica orientada al tiempo y que se use en procesos ágiles. Aplicar las métricas encontrada a un ejemplo concreto, explicando todos los detalles.

Métrica de Tamaño de LDC: Para obtener el valor del tamaño de LDC se debe tomar en cuenta lo siguiente:

- No se cuenta líneas en blanco
- No se cuenta los comentarios
- Se cuenta todo lo demás
- La medida orientada al tamaño no es un método aceptado universalmente.

**· Tamaño = Kilo Líneas de Código
(KLOC)**

9. Finalmente, ¿cuáles son los principales beneficios de hacer métricas para la gestión de un proyecto de software?

Algunos de los beneficios de hacer métricas para la gestión de un proyecto de software son:

- Mayor facilidad de entendimiento común del alcance de la aplicación
- Posibilidad de comparar con una métrica el tamaño funcional de un nuevo sistema con uno existente
- Evaluar con precisión el costo del nuevo software y sus componentes
- Facilita las negociaciones de alcance entre desarrollador y cliente (área funcional)
- Posibilidad de comparar con una métrica el tamaño funcional de un nuevo sistema con uno existente
- Mejor información para mantener el alcance bajo control
- Planificación y presupuesto del soporte técnico de aplicaciones
- Posibilidad de hacer avalúos de activos de software

Bibliografía:

[Nota breve sobre métricas orientadas al tamaño – Acervo Lima](#)

[Estimación de proyectos de software por puntos de función: Introducción - La Oficina de Proyectos de Informática \(pmoinformatica.com\)](#)

[Métricas para la calidad del software - EcuRed](#)

[Métricas del proyecto – Calidad de Software \(wordpress.com\)](#)