**Líneas de Código**

LDC o LOC es una métrica de software que se utiliza para medir el tamaño de un programa, contando el número de líneas del código fuente del mismo. Se suele utilizar para predecir la cantidad de esfuerzo que se requiere para desarrollar un programa. También se usa para estimar la productividad de programación o de mantenimiento una vez que el software es producido.

Hay dos grandes tipos de medidas: líneas de código físicas y líneas de código lógicas. La diferencia entre ambas es que las físicas excluyen las líneas de código comentadas, mientras que las lógicas cuentan todas las sentencias ejecutables.

Ventajas**:**

* Es fácil de medir y se puede realizar de manera automática.
* Es fácil de combinar: muchas otras métricas pueden venir referidas a ella (Ejemplo: "nº de defectos por cada mil líneas de código").
* Ofrece una medida aproximada del tamaño de un software, y de su complejidad.
* Es una medida que existe desde los inicios del software, por lo tanto, existe más información acerca de esta métrica que del resto.

Desventajas**:**

* Mide únicamente la etapa de desarrollo del proyecto, que usualmente es solo el 30-40% de su totalidad.
* Falta de cohesión con la funcionalidad: si bien el esfuerzo está relacionado con la cantidad de líneas de código, no existe una clara correlación entre líneas de código y funcionalidad.
* El número de líneas difiere de una persona a otra.
* No es una medida fiable de productividad personal. El problema es que las métricas están siempre para conocer el proyecto, y no siempre para ver el rendimiento de las personas. Sin embargo, deben obtenerse métricas del estilo: LDC totales por persona, LDC modificadas por persona en un período, LDC nuevas por persona en un período, etc.
* Penaliza a lenguajes de alto nivel. LDC es particularmente ineficaz en la comparación de los programas escritos en diferentes lenguajes a menos que se apliquen factores de ajuste para normalizar idiomas.
* Problemas con múltiples lenguajes: no se recomienda el uso de esta métrica para desarrollos que involucren más de un lenguaje. Para estos casos existe otra métrica llamada puntos de función.
* No existe un estándar acerca del término “línea de código”. ¿Se incluyen los comentarios? ¿Y las definiciones de datos?
* Psicología: un programador cuya productividad se mide en líneas de código tendrá un incentivo para escribir código innecesariamente prolijo y con complejidad innecesaria. Esto ocasiona mayores costos de mantenimiento y mayor esfuerzo para la corrección de errores.

**Métrica orientada a casos de uso**

Puntos de caso de uso es un método de estimación de esfuerzo para proyectos de software, a partir de sus casos de uso.

El método utiliza los actores y casos de uso relevados para calcular el esfuerzo que significará desarrollarlos. A los casos de uso se les asigna una complejidad basada en transacciones y a los actores se les asigna una complejidad basada en su tipo.

Ventajas

* Rápida adaptación a empresas que ya estén utilizando la técnica de Casos de Uso
* Permite cuantificar con bastante precisión las estimaciones de los proyectos de software.
* Permite realizar estimaciones a partir de modelos orientados a objetos.
* Trabaja bien con diferentes tipos de software.
* Da la posibilidad de establecer un tiempo medio de ejecución por punto de casos de uso.
* Es una técnica fácil y rápida de utilizar.

Desventajas

* Todos los casos de uso deben estar redactados
* No existe un estándar para definir casos de uso lo que dificulta la evaluación del método.