

Gestión Desarrollo de Software – 2025

Prof. Politi Raul

Clase 16 (Unidad 5)

Introducción y uso de métricas de software en la planificación

1 - Introducción a las métricas de software

En ingeniería de software, una métrica es una medida cuantitativa que se utiliza para evaluar diferentes aspectos de un proceso, producto o proyecto de software. Su objetivo principal es proporcionar información objetiva y verificable que ayude a tomar decisiones fundamentadas durante la planificación, desarrollo y mantenimiento del sistema.

Dicho de forma simple: 'Lo que no se mide, no se puede mejorar.' Las métricas permiten evaluar el progreso real frente al planificado, detectar problemas temprano y mejorar la calidad del producto.



2 - ¿Por qué son importantes las métricas?

Sin métricas, la gestión de proyectos se basaría en suposiciones o percepciones subjetivas. En cambio, al medir objetivamente el rendimiento del equipo, el avance de las tareas y la calidad del código, los gestores de proyectos pueden planificar con mayor precisión, predecir tiempos y costos, detectar riesgos tempranos y mejorar la productividad.

3 - Tipos de métricas de software

Las métricas se clasifican en tres grandes grupos:

- **Métricas de producto:** miden características del software (tamaño, complejidad, calidad).
- **Métricas de proceso:** evalúan la eficiencia del proceso de desarrollo.
- **Métricas de proyecto:** analizan aspectos de planificación, costo y tiempo.

4 - Métricas más utilizadas en la planificación de software

Entre las métricas más empleadas se destacan:

A) Métricas de tamaño del producto:

- Líneas de Código (LOC): mide el total de líneas escritas. Es fácil de calcular, pero depende del lenguaje.
- Puntos de Función (PF): mide el tamaño funcional del software desde el punto de vista del usuario.

B) Métricas de calidad:

- Número de defectos por módulo, densidad de defectos, porcentaje de casos de prueba exitosos.

C) Métricas de proceso:

- Tiempo promedio para resolver incidencias, velocidad del equipo, retrasos respecto al cronograma.

5 - Cómo se usan las métricas en la planificación

Durante la planificación inicial se usan métricas históricas para estimar esfuerzo, costo y tiempo. Durante la ejecución se comparan métricas reales con las planificadas y al cierre se evalúa el desempeño global.

6 - Ejemplo práctico – Aplicación de métricas en un proyecto

FPA (function points analysis)

Caso: desarrollo de un sistema de ventas online.

1. Se estiman 80 puntos de función.
2. El equipo trabaja a 10 PF por desarrollador por mes.
3. Con 4 desarrolladores → 2 meses.
4. Costo estimado: $80 \text{ PF} \times \text{USD } 40 = \text{USD } 3200$.
5. Durante la ejecución se comparan avances y defectos.
6. Se obtiene productividad y calidad óptimas.

7 - Métricas ágiles

En metodologías ágiles se usan métricas como:

- **Velocidad**: promedio de puntos completados por sprint.
- **Burndown chart**: trabajo pendiente vs. tiempo.
- **Lead Time / Cycle Time**: duración promedio de tareas.
- **Throughput**: número de tareas completadas por período.

8 - Conclusión

Las métricas de software son una herramienta esencial para la gestión de proyectos de TI. Permiten medir objetivamente el desempeño, planificar con precisión y fomentar la mejora continua. Medir, analizar y mejorar son los pilares de una gestión profesional del desarrollo de software.

9 - Desglosando FPA

● Identificar los componentes funcionales

En FPA, se separa **por tipo de funcionalidad**, no por tecnología. Los tipos son:

1. **Entradas Externas (EE)**: datos que el usuario ingresa al sistema.
2. **Salidas Externas (SE)**: información que el sistema entrega al usuario (reportes, resultados, pantallas).
3. **Consultas Externas (CE)**: búsquedas o consultas de información sin modificación de datos.
4. **Archivos Internos (AI)**: tablas o estructuras de datos que mantiene el sistema.
5. **Interfaces Externas (IE)**: datos que se obtienen de otros sistemas o servicios externos.

● Ejemplo de descomposición para un e-commerce

Supongamos tu sistema tiene **usuarios, productos, carrito, pedidos y pagos**:

♦ Entradas Externas (EE)

- Registro de usuario → EE
- Login → EE
- Agregar producto al carrito → EE
- Realizar pedido → EE
- Pago online → EE

♦ Salidas Externas (SE)

- Confirmación de pedido → SE
- Factura o recibo → SE
- Resumen de carrito → SE
- Email de confirmación → SE

♦ Consultas Externas (CE)

- Buscar productos por categoría, nombre o precio → CE
- Consultar estado de pedido → CE

♦ Archivos Internos (AI)

- Tabla Usuarios → AI
- Tabla Productos → AI
- Tabla Pedidos → AI
- Tabla Carritos → AI
- Tabla Pagos → AI

♦ Interfaces Externas (IE)

- Integración con pasarela de pago (MercadoPago, Stripe) → IE
- Integración con sistema de stock externo (si aplica) → IE

● Asignar complejidad a cada elemento

Cada componente se clasifica como:

- **Simple:** pocos campos o reglas simples
- **Medio:** moderada cantidad de campos o validaciones
- **Complejo:** muchos campos, validaciones o relaciones con otros elementos

Ejemplo rápido:

Componente	Tipo	Complejidad	PF aprox.
Registro usuario	EE	Medio	4 PF
Login	EE	Simple	3 PF
Agregar producto al carrito	EE	Medio	4 PF
Buscar productos	CE	Medio	4 PF
Confirmación de pedido	SE	Simple	3 PF
Tabla Productos	AI	Complejo	7 PF
Pasarela de pago	IE	Medio	5 PF

● Sumar PF totales

Después de asignar un PF a cada funcionalidad, se suma todo para obtener el **tamaño funcional del sistema**:

$$PF_{total} = \text{suma de todos los PF de EE} + \text{SE} + \text{CE} + \text{AI} + \text{IE}$$

PFtotal = suma de todos los PF de EE + SE + CE + AI + IE

Ese número será la base para **estimar esfuerzo, tiempo y costo**.

Ejemplo Real de un E-commerce de venta de Ropa.

1. Componentes funcionales y Puntos de Función (PF)

Componente	Tipo	Complejidad	PF
Registro de usuario	EE	Medio	4
Login / autenticación	EE	Simple	3
Agregar producto al carrito	EE	Medio	4
Realizar pedido	EE	Medio	4
Pago online	EE	Medio	4

Confirmación de pedido	SE	Simple	3
Factura / recibo	SE	Medio	4
Resumen de carrito	SE	Simple	3
Email de confirmación	SE	Simple	3
Buscar productos por categoría	CE	Medio	4
Consultar estado de pedido	CE	Medio	4
Tabla Usuarios	AI	Medio	5
Tabla Productos	AI	Complejo	7
Tabla Pedidos	AI	Medio	5
Tabla Carritos	AI	Medio	5
Tabla Pagos	AI	Medio	5
Pasarela de pago (MercadoPago/Stripe)	IE	Medio	5
Integración stock externo	IE	Medio	5

Total PF: 30 PF (ejemplo educativo simplificado)

2. Productividad real del equipo

- **Horas por día:** 8 hs
- **Días por semana:** 5 → 22 días/mes
- **Total horas/mes** = $21,5 \times 5 \approx 107,5$ hs/mes
- **Horas promedio por PF** = 8 hs/PF

PF/mes por desarrollador = $107,5 \div 8 \approx 13$ PF/mes

3. Estimación de tiempo total

2 Estimación del tiempo

$$Tiempo = \frac{PF \text{ totales}}{PF/\text{mes} \times n^{\circ} \text{ de desarrolladores}}$$

- **PF totales** = 30
- **Desarrolladores** = 4
- **PF/mes por dev** = 13

Tiempo = $30 / (13 \times 4) \approx 0,58$ meses $\approx 2,5$ semanas

4. Estimación de costo

- **Costo por hora** = 8 USD
- **Horas por día** = 8 → 64 USD/día
- **Días/mes** = 22 → 1408 USD/mes/dev

Costo total = $4 \times 1408 \times 0,58 \approx 3266$ USD

5. Observaciones

1. Estimación realista basada en horas y tarifa actual.
2. PF totales pueden ajustarse si se agregan funcionalidades adicionales.
3. No incluye testing externo ni deployment, solo desarrollo funcional.
4. Sirve para planificar tiempo, costo y asignación de tareas en proyectos de software.