

## CARRERA DE COMPUTACIÓN

### 1. Datos Informativos

- 1.1. **Módulo:**
- 1.2. **Nivel:** 7 Semestre
- 1.3. **Apellidos y Nombres:** Samira Narvaez - Mendez Estiven – Almer Paguay
- 1.4. **Tema:** Taller: Métricas de Productividad del Desarrollador
- 1.5. **Fecha:** 26/11/2025

### 2. Objetivo

- Comprender qué son las métricas de productividad en desarrollo de software.
- Identificar métricas útiles y riesgosas.
- Reflexionar sobre el uso ético y estratégico de estas métricas.
- Analizar casos reales y construir criterios de aplicación.

### 3. Introducción

Saber si soy productivo como desarrollador va más allá de contar líneas de código. Se trata de preguntarme: ¿el esfuerzo que invertí hoy realmente acercó el proyecto a su meta? Mido mi trabajo para encontrarle el sentido; para detectar esos momentos en los que me atoro y poder simplificarlos, para mejorar cómo hacemos las cosas en equipo y, en el fondo, para tener la tranquilidad de que mi tiempo se traduce en crear algo que funciona bien, es de calidad y resuelve de verdad la necesidad de las personas, no solo en estar ocupado.

#### 4. Contenido

##### 1. Presentación Teórica

Usar las diapositivas:

- Definición de métricas de productividad
- Tipos de métricas (LoC, commits, bugs, velocity, etc.)
- Métricas útiles vs. métricas engañosas
- Buenas prácticas
- Herramientas disponibles

##### 2. Actividad breve dentro de esta sección:

**Reflexión:** ¿Qué métricas ha usado o te han aplicado en prácticas, trabajo o clases?

**Métricas que he usado o visto:**

- En la universidad:
  1. Líneas de código (LoC) → incentivaba código inflado.
  2. Cumplimiento de requisitos → mejor, pero a veces rígido.
- 3. En prácticas/trabajo:
  4. Commits por día → solo muestra actividad, no calidad.
  5. Velocidad (puntos por sprint) → útil para planificar, pero no debe ser meta fija.
  6. Tiempo de resolución de bugs → prioriza rapidez, a veces sacrificando soluciones robustas.
  7. Calidad de código (cobertura, deuda técnica) → objetiva, pero no debe ser obsesión numérica.

##### 3. Análisis de Casos

**Actividad en grupo:**

Entregar 2 casos con distintos enfoques sobre medición de productividad (ej. uno orientado a resultados y otro a control estricto por LoC).

**Preguntas guía:**

- ¿Qué métricas se usan?
- ¿Qué problemas o beneficios identificas?
- ¿Qué mejorarías?

### **Caso 1: Enfoque Orientado a Resultados y Valor (Equipo Ágil)**

Un equipo de desarrollo en una empresa emergente que sigue una metodología ágil como Scrum. El objetivo principal es aprender rápido del mercado y entregar valor tangible a los usuarios en ciclos cortos.

#### **Métricas:**

- Funcionalidades entregadas por sprint.
- Tiempo desde que se empieza una tarea hasta que se despliega (Lead Time).
- Porcentaje de trabajo que genera valor vs. el que es resolver problemas.

#### **Beneficios:**

- El equipo se enfoca en lo que importa: resolver problemas para el usuario.
- Se incentiva la automatización y la eficiencia.

#### **Problema:**

- Puede ser menos tangible para jefes acostumbrados a métricas tradicionales.

#### **Mejora:**

- Combinar con una métrica de calidad (ej., bugs en producción) para evitar descuidar la técnica.

### **Caso 2: Enfoque en Control (Líneas de Código)**

#### **Métricas:**

- Líneas de Código (LoC) escritas por día por programador.
- Horas conectado o en la oficina.
- Número de commits diarios.

#### **Beneficios:**

- Fácil de medir y entender para la gerencia.

#### **Problemas:**

- Incentiva escribir código largo y complejo, no óptimo.
- Castiga las mejores prácticas como refactorizar o simplificar código.
- Crea una cultura de cantidad sobre calidad, muy tóxica.

#### **Mejora:**

- **Eliminarlo.** Es una métrica dañina. Cambiar hacia el enfoque de resultados

#### 4. Dinámica: Construcción de una métrica ideal (20 min)

##### Actividad grupal:

Cada grupo diseña una métrica (o conjunto de métricas) para evaluar la productividad de un desarrollador *en un equipo ágil*.

Debe incluir:

- ¿Qué mide?
- ¿Cómo se calcula?
- ¿Qué problemas evita?
- ¿Cómo se usa éticamente?

#### Preparar presentación para la próxima clase (3 min por grupo)

Link: <https://gamma.app/docs/Dinamica-Construccion-de-tu-Metrica-Agil-Ideal-hynots1tr54lm5l>

#### 5. Conclusiones

- Las métricas son herramientas, no objetivos. Sirven para entender y mejorar procesos, no para vigilar personas.
- Enfócate en el "qué" (resultados/valor), no en el "cómo" (actividad). Métricas como el Lead Time o la calidad son más útiles que las líneas de código.
- La ética es fundamental. Las métricas deben usarse para ayudar al equipo, nunca para castigar o rankear. Transparencia y contexto son clave.
- No hay una solución única. La métrica ideal depende del equipo y proyecto, y se construye con reflexión continua y ajustes.

## 6. Referencia Bibliografía

Sentrio. (2022, August 17). Cómo medir la productividad en desarrollo de software. Sentrio. <https://sentrio.io/blog/medir-productividad-desarrollo-software/>

Valtx. (2023, March 10). Métricas para el desarrollo de software: conoce cuáles son las más relevantes. Valtx. <https://www.valtx.pe/blog/metricas-para-el-desarrollo-de-software-conoce-cuales-son-las-mas-relevantes>

Ortiz, M. G. (2024, October 3). Cómo evaluar y mejorar la productividad en equipos de desarrollo. OpenWebinars.net. <https://openwebinars.net/blog/evaluar-mejorar-productividad-equipos-de-desarrollo/>