



# TAREA 01

## **Estudiantes:**

Nicolas Alexander Herrera Rincon

Julian Alejandro Archila Caro

Stiven Leonardo Sanchez Leon

Santiago Montealegre Romero

Docente: Oscar Eduardo Alvarez Rodriguez

**Fecha de realización:** 05/09/2025

**Fecha de entrega:** 08/09/2025

### **1. ¿Por qué es esencial que los ingenieros de software prioricen el bienestar público?**

Según el Código de Ética, priorizar el bienestar público no es un gesto opcional, sino la base misma de la profesión. El software afecta vidas, privacidad y hasta la seguridad de quienes lo usan, por lo tanto, todo ingeniero debe anteponer el interés colectivo incluso por encima de su propio cliente o empleador. La lógica es clara: un producto que cumple plazos pero compromete la seguridad o la calidad de vida no solo es un mal producto, es una falta ética. Por eso, actuar con responsabilidad frente al público es lo que le da legitimidad y sentido a la ingeniería de software como profesión.

### **2. ¿Qué significa mantener la calidad del producto en software?**

Según el Código de Ética, mantener la calidad del producto significa mucho más que entregar algo que “funcione”. La calidad implica que el software esté bien diseñado, probado y documentado, que respete la privacidad y que no se base en atajos que a futuro comprometan su confiabilidad. Incluso si hay presión de tiempo o costos, el ingeniero tiene la obligación ética de fijar expectativas realistas y no prometer lo que no se puede cumplir. Al final, sacrificar calidad

por velocidad no solo daña la confianza con el cliente, también atenta contra la reputación de la profesión misma.

### **3. ¿Qué implica el compromiso con el aprendizaje continuo?**

El compromiso con el aprendizaje continuo implica aceptar que estamos en un entorno que cambia de forma constante, esto es cierto por nuestra estrecha relación con la tecnología. Un profesional que se estanca deja de cumplir con su deber hacia el cliente y el equipo. Según el Código de Ética, esto no es un lujo, es responsabilidad: actualizarse, corregir errores y buscar mejores formas de trabajar es parte de la obligación de no dañar. Negarse a aprender es tan grave como entregar un producto defectuoso, porque condena al equipo a repetir fallas y al cliente a recibir un servicio cada vez menos competente.

### **4. ¿Qué prácticas promueven la transparencia y la integridad en el desarrollo de software?**

La transparencia y la integridad son simples: decir la verdad y sostenerla con hechos. Eso significa no aprobar un sistema si no está realmente probado, no disfrazar fallos con excusas o ‘features’ y dejar claros los límites de lo que un software puede hacer. También implica reconocer conflictos

de interés y no esconderlos. Si no se llevan a cabo estas prácticas, cualquier decisión técnica pierde legitimidad. En el fondo, la integridad en este campo es no aprovecharse y no jugar con la confianza del cliente.

### **5. ¿Por qué es importante el respeto y apoyo entre colegas en el campo de la ingeniería de software?**

La verdad es que casi nadie trabaja solo en esta profesión. El software suele ser un esfuerzo compartido y cuando falta respeto o apoyo entre colegas, el trabajo se quiebra. Reconocer lo que otros aportan, dar críticas con fundamento y compartir lo que uno sabe no es sólo amabilidad: es asegurar que el equipo tenga cinergia. Mejora la calidad técnica y el proceso de desarrollo.

### **6. Referencias**

[1] D. Gotterbarn, K. Miller, y S. Rogerson, “Software Engineering Code of Ethics,” *Communications of the ACM*, vol. 40, núm. 11, pp. 110–118, Nov. 1997, doi: 10.1145/265684.265699