



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

**Universidad Nacional de Colombia - sede Bogotá**  
**Facultad de Ingeniería**  
**Departamento de Sistemas e Industrial**  
**Curso: Ingeniería de Software 1 (2016701)**

## Tarea 01 - Grupo 7 - UNEmployed

Integrantes:

- ANGEL DAVID GÓMEZ PASTRANA
- DIEGO FELIPE CABRERA SALAMANCA
- PABLO LUNA GUZMÁN
- JAURI ESTEBAN CORTÉS CÁRDENAS

Lectura: <https://ethics.acm.org/code-of-ethics/software-engineering-code/>

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

El código recalca en repetidas ocasiones que el bienestar del público debe ser la prioridad central de la profesión. Esto se debe a que el software no es un producto aislado que funcione como una entidad absoluta, es más bien una herramienta que influye directamente en la vida de las personas, como puede ser el cliente o la empresa, en la forma en que se relacionan con la información y en cómo interactúan con su entorno.

Priorizar el bienestar público significa garantizar que las decisiones técnicas no se tomen únicamente con base en intereses económicos o presiones de tiempo, sino considerando el impacto real que tendrán sobre los usuarios y la sociedad. Esto incluye proteger la privacidad, asegurar la confiabilidad de los sistemas, anticipar riesgos y abstenerse de aprobar software que “disminuya la calidad de vida o dañe el medio ambiente”, como dice el código.

Además, actuar en función del bienestar público fortalece la confianza social en la ingeniería de software como disciplina profesional. Cuando se cumple con este compromiso, demostramos responsabilidad ética, transparencia y respeto hacia la dignidad humana. En últimas, poner al público en el centro no es solo una obligación moral, sino también la condición que asegura que el software sea realmente útil, seguro y justo para todos los involucrados.

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

Mantener la calidad de un producto puede entenderse de distintas formas, según el contexto, los usuarios y las condiciones de entrega. En software, no se trata solo de producir código que cumpla superficialmente con los requerimientos, sino de desarrollar un producto con una estructura clara, un propósito definido y que responda a las necesidades reales del usuario. Esto implica garantizar la seguridad de los datos, diseñar el sistema de

manera que pueda adaptarse a nuevos requerimientos y asegurar su vigencia en el tiempo mediante buena documentación y mantenimiento continuo.

Además, la calidad requiere equilibrar costos razonables con el valor generado, acompañar el desarrollo con pruebas, revisiones y validaciones que confirmen la confiabilidad, seguridad y desempeño del producto. Finalmente, mantener la calidad también significa trabajar bajo estándares profesionales y éticos reconocidos, aplicando buenas prácticas que aseguren tanto la excelencia técnica como la responsabilidad social. Estos elementos —estructura, seguridad, mantenibilidad, documentación, costos, pruebas y ética— son los que garantizan que el software mantenga su calidad a lo largo de todo su ciclo de vida.

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

Implica estar profundizando continuamente en el conocimiento de nuevos desarrollos en cuanto a análisis, especificación, diseño, desarrollo, mantenimiento y testeo de software y su documentación, así como el entendimiento del software, la documentación, el entorno, estándares relevantes y leyes pertenecientes al software y documentos en los que se trabaja.

Además de mejorar la habilidad de crear software de calidad, seguro, confiable, y útil a un precio y en un tiempo razonable, así como de producir documentación acertada, informativa y bien escrita.

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

**Fomentar un entorno ético:** crear una cultura organizacional, es decir, acorde a los valores y organización de cada empresa, donde la integridad y el comportamiento responsable sean valorados.

**Comunicación clara y precisa:** evitar declaraciones falsas, engañosas o especulativas sobre el software, garantizando transparencia en las características y limitaciones del producto.

**Responsabilidad sobre el trabajo:** asumir la detección, corrección y reporte de errores en el software, así como informar de riesgos potenciales para usuarios, el público o el medio ambiente.

**Aprobación Responsable:** Aprobar software sólo si hay una creencia bien fundamentada de que es seguro, cumple con las especificaciones, pasa las pruebas adecuadas, y no disminuye la calidad de vida, la privacidad ni daña el medio ambiente.

**Honestidad sobre Competencias:** Prestar servicio en sus áreas de competencia, siendo *honestos y francos sobre cualquier limitación* de su experiencia y educación.

**Reporte de Problemas:** Identificar, documentar, recopilar evidencia e informar prontamente al cliente o al empleador si, en su opinión, un proyecto probablemente fallará, resultará demasiado costoso, violará la ley de propiedad intelectual o será problemático de otra manera.

**Estimaciones Realistas y Documentación:** Asegurar estimaciones cuantitativas realistas de costo, cronograma, personal, calidad y resultados, y proporcionar una evaluación de la incertidumbre. También, asegurar documentación adecuada, incluyendo problemas significativos descubiertos y soluciones adoptadas

**Cumplimiento legal y ético:** obedecer las leyes aplicables, salvo en casos en que entren en conflicto con el interés público, priorizando siempre la ética profesional.

**Evitar conflictos de interés:** abstenerse de participar en prácticas o asociaciones que comprometan la objetividad profesional.

**Promover el conocimiento público:** contribuir a que la sociedad comprenda la función y el impacto de la ingeniería de software.

**Aprendizaje y mejora continua:** actualizar conocimientos y comprometerse con la aplicación del código de ética en cada proyecto.

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

En primer lugar, crear un ambiente de respeto contribuye a generar confianza y transparencia. Esto permite que los ingenieros puedan manifestar inquietudes éticas, advertir sobre posibles riesgos o reportar incumplimientos del código sin miedo a represalias. De esta forma, no solo se preserva la integridad del proyecto, sino que también se garantiza la responsabilidad frente a los resultados.

Por otro lado, el apoyo entre colegas favorece el crecimiento profesional y la mejora continua. La mentoría, el intercambio de experiencias y el acompañamiento en la adopción de buenas prácticas y medidas de seguridad elevan el estándar de calidad dentro de la profesión.

El respeto se hace evidente también en la dinámica cotidiana: reconocer de manera justa el trabajo ajeno, realizar revisiones objetivas y constructivas, y atender con seriedad las opiniones de los compañeros. Estas actitudes fortalecen tanto la calidad del software como el clima laboral.

Finalmente, evitar obstaculizar de manera injusta la trayectoria de un colega protege su estabilidad y reputación profesional, y con ello se contribuye a que la ingeniería de software mantenga una imagen de seriedad y confianza ante la sociedad.