



UNIVERSIDAD
NACIONAL
DE COLOMBIA

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

¿Por qué es esencial que los ingenieros de software prioricen el bienestar público?

El efecto último de la labor del ingeniero de Software es el bienestar público, y esto es esencial porque de lo contrario, el Software podría ser aprovechado de formas erróneas por gente inescrupulosa; la priorización del bienestar público en el desarrollo de Software, es entonces, una práctica a la que se le pueden atribuir las siguientes virtudes

- **Asegura la integridad y la seguridad del usuario:** Cada parte del proceso de la creación de Software debe estar orientada a su usuario último, quien debe recibir en todos los casos, un software plenamente testado y que cumpla con las especificaciones alrededor de las cuales fue diseñado; es fundamental que el ingeniero reporte cualquier irregularidad a oficiales y personas en posición de acción para asegurar que el Software que el usuario recibirá no empeorará en ninguna manera su calidad de vida o pondrá esta misma ni la integridad del usuario en riesgo.
- **Se compromete con una alocaión de recursos responsable:** El Software debe ser diseñado en mente para ocupar solo y justamente los recursos necesarios para su funcionamiento, asegurando no desperdiciarlos en vano, esto va estrechamente relacionado con asegurar que los usuarios no vean su calidad de vida disminuida de ninguna manera por el uso del Software.
- **Se compromete con el medio ambiente:** A la par que asegura una alocaión y uso responsable de recursos, el ingeniero debe también asegurarse que sus diseños y el Software que aprueba no dañan en ninguna forma el medio ambiente.
- **Privilegia la igualdad:** El ingeniero debe considerar, como parte del desarrollo de cualquier proyecto de Software, las condiciones económicas y ambientales adversas, cualquier impedimento físico, o insuficiencia de recursos, que puedan disminuir de alguna forma el acceso de cierta población al Software que este mismo está creando.
- **Es justo en cada instancia:** El ingeniero debe asegurarse de que su Software es producto de una labor sincera en cada estamento, especialmente en el orientado al público, tomando precaución de lo concerniente al Software, herramientas y documentación.
- **Promueve la enseñanza libre y el acceso a la información:** El ingeniero debe, de forma voluntaria, poner sus conocimientos y habilidades en servicio de la

capacitación de la comunidad, asegurándose de compartir, junto con el conocimiento, el código de ética que implica el desarrollo de Software.

¿Qué significa mantener la calidad del producto en software?

Mantener la calidad del producto en software significa garantizar que el sistema desarrollado siga cumpliendo con los requisitos, expectativas y estándares de calidad a lo largo de todo su ciclo de vida (diseño, construcción, pruebas, despliegue, uso y mantenimiento).

En términos prácticos, implica:

- **Cumplimiento de requisitos funcionales:** El software debe hacer lo que se definió en la especificación: entregar las funciones, procesos y resultados esperados.
- **Cumplimiento de requisitos no funcionales:** Características como rendimiento, seguridad, escalabilidad, usabilidad, confiabilidad y mantenibilidad deben preservarse con el tiempo.
- **Prevención y corrección de defectos:** Detectar y corregir errores oportunamente, evitando que se acumulen fallas que degraden el producto.
- **Adaptación al cambio:** Asegurar que nuevas versiones, actualizaciones o integraciones no afecten negativamente la estabilidad ni la calidad existente.
- **Estandarización y buenas prácticas:** Seguir normas de codificación, pruebas, documentación y gestión de versiones que respalden la calidad de manera constante.
- **Monitoreo y mejora continua:** Evaluar el software en producción (con métricas, retroalimentación de usuarios y pruebas automáticas) y realizar ajustes para mantener o aumentar su nivel de calidad.

¿Qué implica el compromiso con el aprendizaje continuo?

El compromiso con el aprendizaje continuo implica la disposición permanente de una persona para actualizar, mejorar, ampliar sus conocimientos, habilidades y competencias a lo largo de su vida para poder tener mas eficiencia al realizar cualquier actividad, por ejemplo:

- **Adaptarse a los cambios:** sociales, laborales o tecnológicos
- **Buscar nuevas oportunidades de formación:** Cursos, lecturas o experiencias prácticas
- **Reflexionar sobre práctica propia:** identificar todo los aspectos que deba mejorar para reducir los errores al mínimo y tener más eficiencia
- **Mantener la apertura mental:** frente a nuevas ideas y métodos que se vayan presentando al pasar del tiempo

El compromiso con el aprendizaje continuo es asumir la responsabilidad de seguir aprendiendo siempre, para crecer personal y profesionalmente, y responder a las exigencias de un mundo en constante transformación.

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

- **Código claro y documentado:** escribir código entendible, acompañado de comentarios y documentación accesible, permite que otros comprendan y mantengan el sistema con facilidad.
- **Control de versiones:** registrar los cambios de manera ordenada y trazable, mostrando quién hizo qué y cuándo, ayuda a mantener responsabilidad y facilita la revisión del trabajo.
- **Revisiones de código (Code Reviews):** evaluar el trabajo de colegas fomenta la detección temprana de errores, el aprendizaje compartido y asegura que las decisiones técnicas sean justificadas.
- **Buenas prácticas de pruebas (testing):** incluir pruebas unitarias, de integración y de aceptación asegura que el software cumpla lo prometido y minimiza riesgos de fallos ocultos.
- **Gestión ética de datos:** proteger la privacidad de la información sensible y cumplir con normativas (como GDPR o la Ley de Protección de Datos en Colombia) es clave para la confianza del usuario.
- **Comunicación abierta con el equipo y clientes:** informar avances, limitaciones, riesgos o errores sin ocultarlos fortalece la credibilidad.

- **Cumplimiento de estándares y metodologías:** adoptar marcos de trabajo (Scrum, CMMI, ISO/IEC 25010, etc.) ayuda a mantener procesos claros, auditables y transparentes.
- **Evitar el “code smell” y malas prácticas:** no dejar atajos ocultos, puertas traseras o soluciones improvisadas que pongan en riesgo la calidad o la seguridad.

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

El respeto y el apoyo entre colegas en el campo de la ingeniería de software es fundamental por que este trabajo no se desarrolla de manera individual; normalmente se construyen soluciones en equipo integrado distintas perspectivas y habilidades. Algunas razones clave son:

- **Mejora del trabajo en equipo:** el software moderno es complejo y requiere la colaboración de programadores, arquitectos, testers, diseñadores y gestores de proyectos. El respeto favorece la comunicación clara y reduce conflictos innecesarios.
- **Fomenta la innovación:** Cuando las personas se sienten seguras para compartir ideas, se generan soluciones más creativas.
- **Mejora la resolución de problemas:** El apoyo mutuo permite a los equipos superar desafíos complejos de manera más eficiente.
- **Aumenta la productividad:** Un ambiente positivo reduce el estrés y el agotamiento, lo que lleva a un mejor rendimiento.
- **Facilita el aprendizaje:** Los colegas pueden aprender unos de otros, ya sea a través de la revisión de código o la tutoría.