

# Tarea 1

## Ingeniería de Software I

### 2025 - 1

Juan Jeronimo Gomez Rubiano (jujgomezru@unal.edu.co)  
Cristian Leonardo Castañeda Olarte (crcastanedao@unal.edu.co)  
David Santiago Velasquez Gomez (dvelasquezg@unal.edu.co)  
Alan Ryan Cheyne Gomez (acheyne@unal.edu.co)  
**Lector:** Oscar Eduardo Alvarez Rodriguez (oalvarezr@unal.edu.co)

## Pregunta 1

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

La aplicación de un código ético, en general, en la Ingeniería de Software, está íntimamente relacionado con la manera en la que los computadores, y las herramientas de software, se están volviendo cada vez más prevalentes en la sociedad moderna. Ciertamente, este es un enfoque más práctico que purista, y la ética deontológica clásica probablemente considere otros imperativos como los más adecuados para construir un código ético. No obstante, más allá de justificarlo, el nivel de relevancia que están adquiriendo las herramientas de software recalca la influencia real que tienen los Ingenieros de Software en prácticamente cualquier área del conocimiento, y por tanto, lo fundamental que es exigir que sigan un código ético coherente y completo.

El ámbito del bienestar público es importante al punto que ha sido designado como el Punto 1 del código de ética y práctica profesional. Esto es porque las herramientas de software deberían tener en su núcleo una labor social, y la conducta y práctica del Ingeniero debe estar enfocada a los mismos estándares. Uno de los aspectos más importantes es modular los intereses personales que pueda tener el ingeniero a la hora de desarrollar un producto. No debe dejar que sus preferencias impacten el carácter social de sus proyectos. Al mismo tiempo, también tiene la obligación de no mantener el silencio, si el proyecto en el que está trabajando presenta riesgos significativos a, por ejemplo, la seguridad de las personas, el bienestar, el medio ambiente, o la calidad de vida. Si el software en el que está trabajando presenta serios problemas técnicos de calidad, o se considera que puede ser usado con intenciones nefarias, o de manera inequitativa (incluyendo también problemas de accesibilidad), el Ingeniero debe manifestar sus preocupaciones, así como proponer medidas correctivas para estos riesgos. Todo esto debe ser juzgado desde un punto de vista objetivo e informado, con miras de ofrecer su particular punto de vista profesional, como algo que se debe considerar, si se considera que la consecuencia de un producto puede ser

perjudicial para una o varias personas.

## Pregunta 2

¿Qué significa mantener la calidad del producto en software?

Mantener la calidad del producto en software significa que el trabajo realizado en las distintas fases de Ingeniería de Software debe realizarse con los más altos estándares profesionales posibles. Esto implica poner el mayor empeño posible en los distintos aspectos del proceso de desarrollo, para asegurarse que el producto final sea lo más funcional y eficiente posible, velando por la seguridad y satisfacción de los usuarios.

Algunos de los aspectos que toca resaltar durante el proceso de desarrollo de calidad son:

### Desarrollo

Asegurar que el producto final sea de alta calidad. También asegurar que el cobro y presupuesto asignado sean razonables, es decir, coherentes con las ganancias esperadas del producto, así como asegurar que el producto sea asequible al público, en la medida de lo que sea conveniente para que sea sostenible, y pueda ser usado sin vulnerar la información sensible que pueda manejar.

### Contexto social

Un proyecto de calidad es aquel que responda a necesidades claras en un entorno socio-económico, y no uno que existe en el abstracto para resolver necesidades individuales, o incoherentes a las demandas de la sociedad. Igualmente, como se mencionó en el punto 1, un Ingeniero de Software no debe realizar productos con fin de obtener un beneficio económico propio o de terceros, a expensas de los usuarios, o poblaciones vulnerables que pudieran en otros casos, verse beneficiadas o depender de las herramientas de software.

### Responsabilidad individual

El desarrollador debe poder confirmar que él, ella, o su equipo, tenga las habilidades requeridas para construir un proyecto de calidad, o admitir honestamente cuando el proyecto propuesto se sale de su área de pericia, y declinar ofertas si no es capaz de garantizar los objetivos de calidad del proyecto. Igualmente, deben asegurarse que los objetivos generales y específicos definidos para la ejecución del proyecto sean realistas y pertinentes, para poder asegurar un desarrollo constante y productivo.

### Metodología

El desarrollador, y el equipo en conjunto, deben velar, además de los conocimientos y capacidades para programar, y desarrollar software, por el uso de metodologías adecuadas de trabajo en equipo para coordinar la construcción de un proyecto de manera eficiente. De igual manera, deben asegurarse de entender precisamente las especificaciones de cada proyecto, para solventar las dificultades que vayan surgiendo en el proceso de desarrollo, y entregar un producto final acorde a los requerimientos del usuario.

## Documentación

Durante el proceso de desarrollo de software, el desarrollador debe asegurarse de incluir documentación extensiva, y accesible, sobre los distintos componentes y funcionalidades del proyecto. De esta manera, aseguran que requerimientos de uso sean fáciles de entender, los distintos bugs y problemas que reporten los usuarios puedan ser atendidos de manera autónoma, o incluso, usar la documentación para realizar el debugging y mantenimiento de forma más eficiente, pues se entiende de manera más exacta, el código que fue usado, y las modificaciones necesarias para resolver problemas de usuario.

## Testing

En las etapas finales de desarrollo, los ingenieros de software deben tener herramientas profesionales de pruebas de software. El uso adecuado de estas herramientas, y la implementación de protocolos profesionales, asegura que los distintos errores, o problemas de diseño que pueda tener el software durante la etapa de desarrollo, sean identificados de manera oportuna, asegurando por tanto que todos estos inconvenientes sean corregidos antes del despliegue definitivo del proyecto.

## Mantenimiento

El Ingeniero de Software debe enfrentar las tareas de mantenimiento de software con la misma rigurosidad profesional, que la aplicada a las etapas anteriores de desarrollo. Esto implica una revisión constante del código posterior al despliegue, o asignar un equipo de desarrolladores competente para esta labor. Todo producto de Ingeniería de Software debe estar sujeto a escrutinio y evaluación, para garantizar la satisfacción del usuario a largo plazo.

## Responsabilidad legal

El Ingeniero de Software tiene responsabilidad frente al uso legal de herramientas o documentos, para el desarrollo de sus productos. Debe asegurarse de que cualquier código o archivo usado para el desarrollo (por ejemplo, entrenamiento de Modelos de Inteligencia Artificial Generativo), sean de uso legal y justo. Un buen Ingeniero de Software no puede utilizar recursos externos sin consentimiento, o reconocimiento a los autores originales, ni usar dichos recursos de maneras no autorizadas, para un beneficio económico o profesional a costa de otros desarrolladores y profesionales. También debe velar por la seguridad y privacidad de la información manejada por sus productos. Hoy en día, prácticamente todas las industrias a nivel mundial utilizan software por cuestiones de eficiencia y productividad, lo que incluye bancos, correo electrónico, ofimática, comunicaciones, entre muchas otras industrias que necesitan garantizar la seguridad y privacidad de sus usuarios para poder brindar un servicio de calidad, tarea en la cual el compromiso y la habilidad del Ingeniero de Software es fundamental.

## Pregunta 3

¿Qué implica el compromiso con el aprendizaje continuo?

El compromiso con el aprendizaje continuo está principalmente relacionado en el **Principio 8: YO MISMO (SELF)**. Este principio establece que "Los ingenieros de software deberán participar en el aprendizaje permanente sobre la práctica de su profesión y deberán promover un enfoque ético en la práctica de la profesión".

Esto implica las siguientes acciones y responsabilidades para un ingeniero de software:

1. **Actualizar el Conocimiento Técnico:** Seguir aprendiendo y mejorando el conocimiento sobre los avances en análisis, especificación, diseño, desarrollo, mantenimiento, pruebas y documentación de software.
2. **Mejorar Habilidades Prácticas:** Esforzarse continuamente por mejorar la capacidad de crear software seguro, confiable, útil y de alta calidad de manera económica.
3. **Mejorar Habilidades de Documentación:** Trabajar en la habilidad para producir documentación precisa, informativa y bien redactada.
4. **Ampliar la Comprensión:** Mejorar la comprensión no solo de los aspectos técnicos de la ingeniería de software, sino también de su gestión, los estándares relevantes y el entorno en el que opera el software.
5. **Conocer el Marco Legal y Regulatorio:** Mantenerse informado y mejorar el conocimiento de las leyes y regulaciones pertinentes que afectan su trabajo.
6. **Profundizar en la Ética Profesional:** Mejorar el conocimiento del propio Código de Ética, sus interpretaciones y cuestiones éticas relacionadas con la profesión.

En otras palabras, el compromiso con el aprendizaje continuo significa que los ingenieros de software tienen la obligación profesional de mantenerse actualizados en todos los aspectos relevantes de su campo (técnicos, de gestión, legales, éticos) para mejorar constantemente sus habilidades y la calidad de su trabajo a lo largo de toda su carrera.

## Pregunta 4

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

No existe un conjunto de prácticas preestablecidas para promover la transparencia y la integridad en el desarrollo de software que los profesionales puedan seguir como si se tratara de un libro de cocina. En su lugar, cada profesional de la ingeniería de software debe regirse bajo, además de las leyes que lo gobiernen, por un conjunto de principios éticos que guíen su actuación de su carrera profesional. De esta necesidad surge la importancia de implementar códigos de conducta como el que proporciona *The Software Engineering Code of Ethics and Professional Practice* [1], cuyo objetivo es que cada ingeniero de software se comprometa con una serie de principios éticos orientados a la promoción del interés social, la transparencia y la integridad en el desarrollo de software, los cuales brindarán un marco de trabajo suficientemente flexible para que el desarrollador pueda actuar de manera crítica y coherente con dichos principios.

De este modo, tanto organizaciones como profesionales pueden asegurar que su labor dentro de la industria está orientada al desarrollo de software íntegro, transparente y en favor del bienestar social y la mejora de calidad de vida.

## Pregunta 5

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

En primer lugar, porque el desarrollo de software suele ser un área en el que el trabajo en equipo se distribuye a lo largo de varias fases que van desde el entender las necesidades del cliente, liderar al equipo, desarrollar y construir el código, diseñar interfaces y experiencias de usuario y análisis, pruebas y mantenimiento del mismo. Todas estas actividades deben de distribuirse a varios individuos que deben de tener una comunicación constante y preferiblemente cordial. Orientándonos un poco más hacia una perspectiva del talento humano es necesario entender que un entorno sin respeto entre los empleados puede dar origen a malentendidos o especificaciones ambiguas, lo que a su vez puede llevar a generar horas de trabajo innecesario mientras que un equipo que se comunica abiertamente puede discutir ideas y prevenir conflictos innecesarios por lo que es fácil concluir que un equipo con una comunicación adecuada no solo entrega software de calidad sino que también constituyen una fuerza laboral avanza más rápido y se adapta mejor a los desafíos.

Ahora, en segundo lugar, desde un punto de vista basado en la colaboración externa, es decir, no un equipo trabajando hombro a hombro sino sencillamente en la colaboración altruista como lo son: los foros de programación, tutoriales en video, revisar el código de un compañero o si ahondamos en comunidades más dedicadas a la idea del beneficio colectivo sobre el lucro individual, es posible encontrar grupos enfocados en Linux, Python o GODOT solo por mencionar algunos, en los que el principio del Open Source está enfocado en mejorar herramientas que otros puedan usar y desarrollar más a fondo sin ningún costo monetario, citando una de las grandes frases que deja la ciencia sobre la importancia de la accesibilidad de los conceptos previos: “Si he visto más lejos es porque estoy sentado sobre los hombros de gigantes” (Isaac Newton). Los ejemplos antes mencionados son un ejercicio que ayuda a entender que el trabajo en el campo de la ingeniería de software tiene un propósito más allá del código entregable, también es un catalizador para mejorar y potenciar otros sistemas.

## Referencias

- [1] ACM and IEEE-CS, "Software Engineering Code of Ethics and Professional Practice," Version 5.2, 1999. [Online]. Disponible en: <https://www.acm.org/code-of-ethics/software-engineering-code>