



Universidad de las Fuerzas Armadas ESPE

Departamento: Ciencias de la computación

Carrera: Ingeniería en Tecnología de la Información

Taller académico N°: 2

1. Información General

- **Asignatura:** Metodología de Desarrollo de Software
 - **Apellidos y nombres de los estudiantes:** Cristhian Marcalla, Alexandro Molina, Kevin Cañola
 - **NRC:** 20967
 - **Fecha de realización:** 30/04/2025
-

2. Objetivo del Taller y Desarrollo

Objetivo del Taller:

Analizar los principios fundamentales, retos actuales, implicaciones éticas y evolución de la ingeniería de software profesional, a través del estudio de distintas preguntas clave, con el fin de comprender su impacto en el desarrollo y uso de sistemas informáticos en el siglo XXI.

Desarrollo:

EJERCICIOS

1.1. Explique por qué el software profesional no sólo son programas que se desarrollan para un cliente.

El software especializado no se restringe solo a programas creados para un cliente particular, ya que abarca una extensa gama de productos diseñados para varios usuarios, plataformas y contextos. Este software puede incluir desde sistemas operativos, herramientas de desarrollo, programas embebidos, hasta aplicaciones comerciales de tipo genérico. Además, la creación de software profesional incluye tareas como administración de proyectos, evaluaciones de calidad, documentación, mantenimiento, soporte y mejoras constantes, lo que trasciende el mero desarrollo de código. Su atención se centra en la calidad, el reaprovechamiento, la escalabilidad y la observancia de normas técnicas y éticas.



1.2. ¿Cuál es la principal diferencia entre desarrollo de software genéricos y desarrollo de software personalizado? ¿Qué significa esto en la práctica para los usuarios de productos de software genérico?

El software genérico y el personalizado se distinguen fundamentalmente por su público objetivo y su grado de adaptabilidad. El software genérico, concebido para una amplia gama de usuarios, impone a menudo una adaptación de los procesos preexistentes a su estructura, lo que puede conllevar limitaciones funcionales o la incorporación de elementos superfluos. En cambio, el software personalizado se desarrolla específicamente para satisfacer los requerimientos singulares de un cliente, ofreciendo una alineación precisa con sus necesidades operativas.

1.3. ¿Cuáles son los cuatro atributos importantes que debe tener todo software profesional? Sugiera otros cuatro atributos que en ocasiones sean significativos.

Las cuatro características fundamentales de cualquier software profesional son:

- Seguridad: El software debe funcionar de manera exacta y prevista en todas las circunstancias anticipadas.
- eficacia: Es necesario aprovechar al máximo los recursos del sistema, tales como memoria y tiempo de proceso.
- Protección: Es necesario resguardar los datos del usuario y prevenir ingresos no permitidos.
- Conservación: Debe ser sencillo de modificar para rectificar fallos, ajustarse a nuevos requisitos o optimizar su desempeño.

Otros cuatro rasgos que pueden ser relevantes dependiendo del contexto son:

- Escalable: Habilidad del software para expandirse y ajustarse a cargas laborales más grandes o a más usuarios.
- Manejabilidad: Pronta implementación del software en diversos sistemas operativos o plataformas.
- Aplicabilidad: Conexión y uso sencillo para el usuario final.
- Cantidad de disponibles: Habilidad del software para mantenerse en funcionamiento durante el tiempo necesario, particularmente en sistemas esenciales.

1.4. Además de los retos de la heterogeneidad, cambio empresarial y social, y confianza y seguridad, identifique otros problemas y retos que sea probable que enfrente la ingeniería de software en el siglo XXI. (Sugerencia: piense en el ambiente).

Consumo Energético: El software demanda mucha energía en su operación y la de su infraestructura, generando emisiones.

Huella de Carbono del Desarrollo: La creación de software también produce emisiones a través de diversos procesos.



Impacto Sectorial: Un software mal diseñado puede llevar a un uso ineficiente de recursos en otros sectores clave.

Desechos Electrónicos: El software puede acelerar la obsolescencia del hardware, contribuyendo a la basura electrónica.

1.5. Con base en su conocimiento de algunos tipos de aplicación estudiados en la sección 1.1.2, explique, con ejemplos, por qué diferentes tipos de aplicación requieren técnicas especializadas de ingeniería de software, para apoyar su diseño y desarrollo.

Las distintas categorías de aplicaciones exigen enfoques especializados en ingeniería de software debido a sus características particulares y requisitos específicos. Por ejemplo, los sistemas integrados requieren tiempos de respuesta en tiempo real y una alta confiabilidad operativa, mientras que las aplicaciones para empresas necesitan capacidad de ampliación e integración con múltiples plataformas. Las aplicaciones móviles se centran en la eficiencia de recursos y una experiencia de usuario optimizada, mientras que los sistemas críticos deben someterse a rigurosos procesos de validación y verificación para garantizar la ausencia de fallos que puedan comprometer la seguridad o la funcionalidad esencial. Por lo tanto, se necesita utilizar diversas técnicas, enfocándose en el problema central que se busca solucionar, y así optimizar el tiempo y los recursos para la elaboración de dicho programa.

1.6. Explique por qué existen ideas fundamentales de la ingeniería de software que se aplican a todos los tipos de sistemas de software.

Las ideas centrales de la ingeniería de programas son reglas que se usan con todos los tipos de sistemas. Esto es porque tocan problemas comunes al hacer software como ser complicado, tener que arreglar fallas, asegurar que este bien hecho y trabajar en equipo. Estas reglas ayudan a crear sistemas más claros, duraderos y fáciles para cambiar a nuevas exigencias sin importar su clase o tamaño.

1.7. Explique cómo el uso universal de la Web cambió los sistemas de software.

La expansión de la Web cambió por completo la forma en que se desarrollan los sistemas de software. Se pasó de modelos centralizados a arquitecturas distribuidas, capaces de atender a millones de usuarios al mismo tiempo. Esto también trajo la necesidad de crear interfaces accesibles desde navegadores, fortalecer la seguridad, y actualizar el software de forma continua. Además, fomentó el diseño que funciona en diferentes dispositivos y el uso de servicios web para que distintos sistemas puedan comunicarse entre sí. Por lo tanto, podemos decir que la web hizo que el software fuera más flexible, escalable, accesible y adecuado para un mundo cada vez más conectado.

1.8. Analice el hecho de si los ingenieros profesionales deben ser certificados en la misma forma que los médicos o abogados.

La certificación profesional de ingenieros, parecida a la pedida en trabajos como medicina o ley, es un asunto que se habla. En Ecuador, la Ley de Trabajo de Ingeniería dice que los



ingenieros deben anotar su título y, en algunos casos, revisar títulos de otros países para trabajar legalmente. Sin embargo, no se pide una certificación necesaria. A nivel mundial, hay modelos como el de licencia para trabajar, que necesita pasar pruebas y seguir requisitos extras, y el de certificación profesional, que chequea habilidades mediante exámenes sobre saberes, vivencias y actitud. La aplicación de una certificación necesaria para ingenieros podría mejorar la calidad y el compromiso en el trabajo profesional aunque plantearía dificultades en cuanto a precios, facilidad y adaptación al entorno local.

1.9. Para cada una de las cláusulas del Código de ética ACM/IEEE que se muestra en la figura 1.3, sugiera un ejemplo adecuado que ilustre dicha cláusula.

PÚBLICO

Un desarrollador de una aplicación de mensajería identifica la existencia de una vulnerabilidad que puede exponer datos personales de los usuarios y, bajo el principio de «dar primero la noticia», se comunica con seguridad del organismo regulador para que se tomen las medidas necesarias para proteger el interés de la sociedad en la que se trabaja.

CLIENTE Y EMPLEADOR

Un ingeniero recibe el encargo de su empresa para recopilar y vender estadísticas de uso de una app sin informar a los usuarios; él decidirá añadir su visión de la mejor práctica al crear exactamente una ventana de consentimiento que garantice el mejor interés del cliente y del público.

PRODUCTO

Un equipo que diseña un sistema de pagos realiza pruebas en cada uno de los módulos, prueba automática y manual, y entrega el software al cliente solo cuando ha verificado que se está cumpliendo con lo dispuesto en el PCI DSS.

JUICIO

Un programador, sujeto a una presión manifiesta de completar antes de lo planeado determinado proyecto, hace valer su independencia profesional y prosigue con la fase de pruebas de los módulos y la integración de estos, haciendo también valer su juicio sobre la calidad de este.

GESTIÓN

Un jefe de proyecto introduce revisiones de ética periódicas en el desarrollo de un software médico, alimentando el desarrollo de políticas claras de calidad y de seguridad dentro del marco de todo el ciclo de vida.

PROFESIÓN



Un ingeniero da su contribución mediante la creación de código abierto para una biblioteca de cifrado y organiza talleres gratuitos sobre buenas prácticas que transmite a la comunidad para proteger la reputación profesional colectiva.

COLEGAS

Una desarrolladora brinda mentoría a un compañero que se incorpora a la empresa participando en la revisión de su código con respeto y sugiriendo mejoras que favorecen su formación y el trabajo en equipo.

UNO MISMO

Un profesional se da tiempo cada mes para formarse en nuevas metodologías ágiles y ciberseguridad, asegurando así su aprendizaje permanente y su ética profesional.

1.10. Para ayudar a contrarrestar el terrorismo, muchos países planean o desarrollan sistemas de cómputo que siguen la pista a gran cantidad de sus ciudadanos y sus acciones. Claramente esto tiene implicaciones en cuanto a la privacidad. Discuta la ética de trabajar en el desarrollo de este tipo de sistema.

El avance en sistemas de supervisión para luchar contra el terrorismo presenta un importante dilema ético. Aunque pueden incrementar la seguridad, también tienen el potencial de infringir la privacidad y los derechos básicos de los ciudadanos. Conforme al código de ética de la ACM/IEEE, los ingenieros deben dar prioridad al bienestar de la población, respetar la privacidad y comportarse con responsabilidad. Trabajar en estos sistemas solo es moralmente correcto si existen claridad, límites legales precisos, defensa frente a abusos y una supervisión apropiada. Por otro lado, pueden transformarse en instrumentos de control abusivo y represión.

3. Referencias (Norma APA 7.0)

- Gotterbarn, D., Miller, K., & Rogerson, S. (2018). *ACM Code of Ethics and Professional Conduct*. Association for Computing Machinery.
<https://www.acm.org/code-of-ethics>
- Zuboff, S. (2019). *The age of surveillance capitalism*. PublicAffairs.
- Richards, N. M. (2013). The dangers of surveillance. *Harvard Law Review*, 126(7), 1934–1965.
- Sommerville, I. (2011). *Ingeniería de software* (9ª ed.). Pearson Educación.
- Pressman, R. S., & Maxim, B. R. (2014). *Ingeniería de software: Un enfoque práctico* (8ª ed.). McGraw-Hill.
- Sommerville, I. (2016). *Ingeniería del software* (10ª ed.). Pearson Educación.
- • Sommerville, I. (2011). *Ingeniería de Software* (9ª ed.). Pearson Educación.
- • Pressman, R. S. (2010). *Ingeniería del Software: Un enfoque práctico* (7ª ed.). McGraw-Hill.



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



- IEEE Computer Society. (2020). *Software Engineering Body of Knowledge (SWEBOK) V3.0*. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
 - Gotterbarn, D., Miller, K., & Rogerson, S. (1997). *Software Engineering Code of Ethics and Professional Practice* (Versión 5.2). Association for Computing Machinery (ACM) y IEEE Computer Society. <https://www.acm.org/code-of-ethics/software-engineering-code>
 - Sommerville, I. (2011). *Software engineering* (9th ed.). Boston: Addison-Wesley.
 - Pressman, R. S., & Maxim, B. R. (2014). *Software engineering: A practitioner's approach* (8th ed.). New York: McGraw-Hill Education.
-