



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



1. DATOS INFORMATIVOS

CARRERA: Tecnologías de la Información

ASIGNATURA: *Metodología de desarrollo de Software*

TEMA: Fundamentos de la Ingeniería
de Software

DOCENTE: Ruiz Robalino Jenny Alexandra

Integrantes: Nayely Simbaña

Sebastian Medina

Benjamín López

FECHA: 13/10/2025 **NRC:** 29022

2. DESARROLLO

Autoevaluación del capítulo 1

1. La persona que acuñó por primera vez el término “ingeniería del software” fue:

- Margaret Hamilton.
- Margaret Sanger.
- Margaret Atwood.

Justificación: Margaret Hamilton, ingeniera del MIT, fue quien utilizó por primera vez el término *software engineering* durante el desarrollo del software del Apolo 11 de la NASA. (Somerville, 2011, p. 6).

2. Los elementos que componen el software son:

- Personal, proceso y producto.
- Programas, procedimientos, documentación y datos relacionados.
- Programas o instrucciones, partes y piezas y datos.

Justificación: El software incluye programas ejecutables, la documentación que los describe y los datos que utilizan. No se limita al código fuente. (Ruiz, 2018).

3. Oficialmente, el término ingeniería del software se acuñó en:

- La Conferencia de la OTAN de 1968.
- La Conferencia de la CEPAL de 1963.
- La Conferencia de la OTAN de 1986.

Justificación: En la Conferencia de la OTAN sobre Ingeniería del Software realizada en Garmisch, Alemania, en 1968, se introdujo oficialmente el término Software Engineering para describir la necesidad de aplicar principios de ingeniería al desarrollo de software. Este evento fue una respuesta directa a la llamada “crisis del software”, caracterizada por sobrecostos, retrasos y fallas en los sistemas desarrollados. (Somerville, 2011, p. 7).

4. La definición de tipo de software correcto es:

- Programas que resuelven necesidades específicas de las organizaciones (software de sistemas).
- Conjunto de programas que han sido escritos para servir a otros programas (software de gestión o aplicación).
- Software que hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo (software de inteligencia artificial).

Justificación: El software de inteligencia artificial emplea algoritmos no numéricos para resolver problemas complejos mediante razonamiento y aprendizaje, simulando la inteligencia humana. Se diferencia del software de sistemas y de aplicación por su capacidad de adaptación y análisis. (Ruiz, 2018).

5. ¿Cuáles son los atributos de un buen software?

- Funcionalidad y el rendimiento requerido por el usuario.
- Hacer que se malgasten los recursos del sistema.
- Mantenable, confiable y fácil de utilizar.

Justificación: Somerville (2011) define que un software de calidad debe poseer atributos esenciales como: mantenibilidad, que implica facilidad de modificación y mejora; confiabilidad, que garantiza que el software funcione sin fallos bajo condiciones específicas; y usabilidad, que permite que los usuarios puedan interactuar con él de manera intuitiva y eficaz. Estos atributos son fundamentales para lograr la satisfacción del cliente y la sostenibilidad del producto a largo plazo. (Somerville, 2011, p. 12).

6. Las características del software son:

- El software usa componentes estándar con funciones e interfaces bien definidas.
- El software se desarrolla o modifica con intelecto, no se fabrica en el sentido clásico.
- El software se desgasta con el transcurso del tiempo.

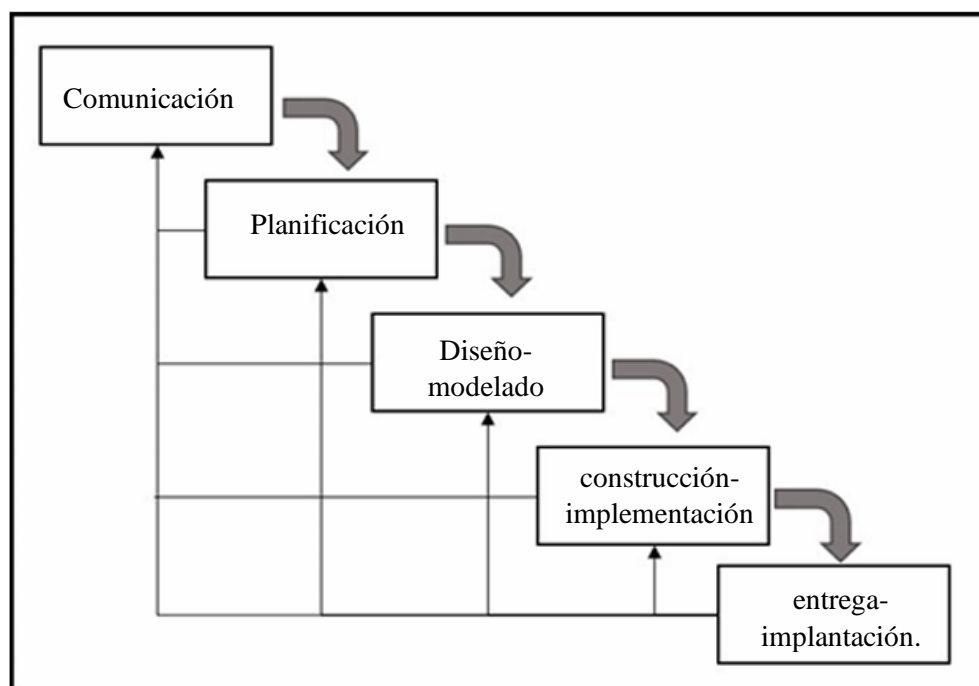
Justificación: A diferencia de los productos físicos, el software no se ensambla ni se fabrica en una línea de producción. Su desarrollo es un proceso intelectual y creativo que involucra análisis, diseño, codificación y pruebas. Ruiz (2018) señala que cada versión del software puede modificarse o ampliarse sin que exista una “producción” material, y los errores tienden a surgir por fallas humanas, no por desgaste físico. (Ruiz, 2018, p. 16).

7. La crisis del software se refiere a los problemas que desde sus inicios ha ido experimentado este. Muchas veces los problemas de gran magnitud se generan debido a la mínima eficacia que presenta una gran cantidad de empresas al momento de realizar un software.

- Verdadero.
- Falso.
- Ninguna de las opciones.

Justificación: La llamada crisis del software se refiere a los problemas que las organizaciones comenzaron a enfrentar al desarrollar programas cada vez más grandes y complejos: sobrecostos, incumplimiento de plazos, errores y sistemas inestables. Somerville (2011) explica que esta crisis motivó la creación de métodos más estructurados de desarrollo, dando origen a la ingeniería del software como disciplina formal. (Somerville, 2011, p. 8).

8. A partir del siguiente gráfico, los nombres de las fases del modelo en Cascada (Waterfall) son:



Justificación: El modelo en cascada o Waterfall es uno de los modelos de proceso más tradicionales. Propone una secuencia ordenada de fases donde cada una depende del resultado de la anterior: primero se comunican los requisitos, luego se planifica el proyecto, se diseña la solución, se construye e implementa el sistema y, finalmente, se entrega o implanta al usuario. Ruiz (2018) señala que este modelo facilita el control del avance, aunque tiene poca flexibilidad ante cambios. (Ruiz, 2018, p. 25).

9. El modelo de proceso de software en espiral propuesto por Bohem conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. La etapa que no pertenece al modelo es:

- Evaluación del cliente.
- Comunicación con el cliente.
- Definición de un paradigma de desarrollo.

Justificación: El modelo espiral combina características iterativas y de gestión de riesgos. Sus fases son: planificación, análisis de riesgos, ingeniería (desarrollo) y evaluación del cliente. Cada ciclo produce una versión más completa del software. La opción “definición de un paradigma de desarrollo” no forma parte de este proceso, ya que el modelo espiral es en sí mismo un paradigma. (Somerville, 2011, p. 37).

10. Se construye un buen sistema de información considerando que el punto de partida es:

- La definición de requisitos claros es parte del proceso, pero no es del todo importante.
- Utilizar un proceso definido con fases claras, donde cada una de estas genera un producto final.
- Utilizar herramientas de desarrollo como medio para alcanzar un producto de calidad.

Justificación: La creación de sistemas de información exitosos requiere aplicar procesos definidos y estructurados, que incluyan fases con objetivos, actividades y productos intermedios bien determinados. Ruiz (2018) enfatiza que esto garantiza la trazabilidad, la calidad y la gestión eficiente del proyecto. Somerville (2011) coincide en que un proceso ordenado reduce riesgos y mejora la satisfacción del cliente. (Ruiz, 2018, p. 29; Somerville, 2011, p. 14).

3. CONCLUSIONES:

La ingeniería del software constituye una disciplina fundamental que permite desarrollar sistemas de calidad mediante procesos estructurados, controlados y orientados a resultados. Su origen, motivado por la crisis del software, impulsó la aplicación de principios de ingeniería para mejorar la eficiencia, la confiabilidad y el mantenimiento de los programas. Comprender sus conceptos básicos, los modelos de desarrollo y los

atributos de un buen software garantiza la creación de soluciones tecnológicas más seguras, sostenibles y alineadas con las necesidades del usuario.

4. RECOMENDACIONES:

Se recomienda aplicar de manera constante los principios y metodologías de la ingeniería del software en cada etapa del desarrollo, fomentando la planificación, la documentación y la evaluación continua. Esto permitirá obtener productos de calidad, reducir errores, optimizar recursos y asegurar que las soluciones tecnológicas respondan eficazmente a las necesidades reales de los usuarios y las organizaciones.

5. Referencias:

Ruiz Robalino, J. A., & Lucio Moreno, X. I. (2025). Fundamentos de ingeniería de software. Editorial Universidad de las Fuerzas Armadas ESPE.

Somerville, I. (2011). Ingeniería del software (9.^a ed.). Pearson Educación.