

1. DATOS INFORMATIVOS

Carrera: Ing. En tecnologías de la información

Asignatura: Metodología de la investigación de software

Tema del taller: Autoevaluación Capítulo 1

Docente: Ing. Jenny Ruiz

Integrantes: César Herrera, Matias Intriago, Kelly Montalvo

Fecha: 13/10/2025 Paralelo: 29022

2. DESARROLLO

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

Margaret Hamilton.

Margaret Sanger.

Margaret Atwood.

Justificación: Ella fue quien usó por primera vez el término “ingeniería del software” mientras trabajaba en la NASA, para destacar la importancia del desarrollo de programas como una disciplina formal.

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 no solo son programas, también incluye la documentación, los datos y las instrucciones necesarias para su correcto funcionamiento.

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 esta conferencia se habló por primera vez de “ingeniería del software” para resolver los problemas de calidad y organización que tenía la industria.

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: Este tipo de software utiliza algoritmos inteligentes para resolver problemas complejos que no pueden resolverse con métodos tradicionales.

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.

Mantenible, confiable y fácil de utilizar.

Justificación: Un buen software debe ser sencillo de mantener, confiable en su funcionamiento y práctico para el usuario.

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 se crea con ideas, análisis y programación, no en una línea de producción

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.

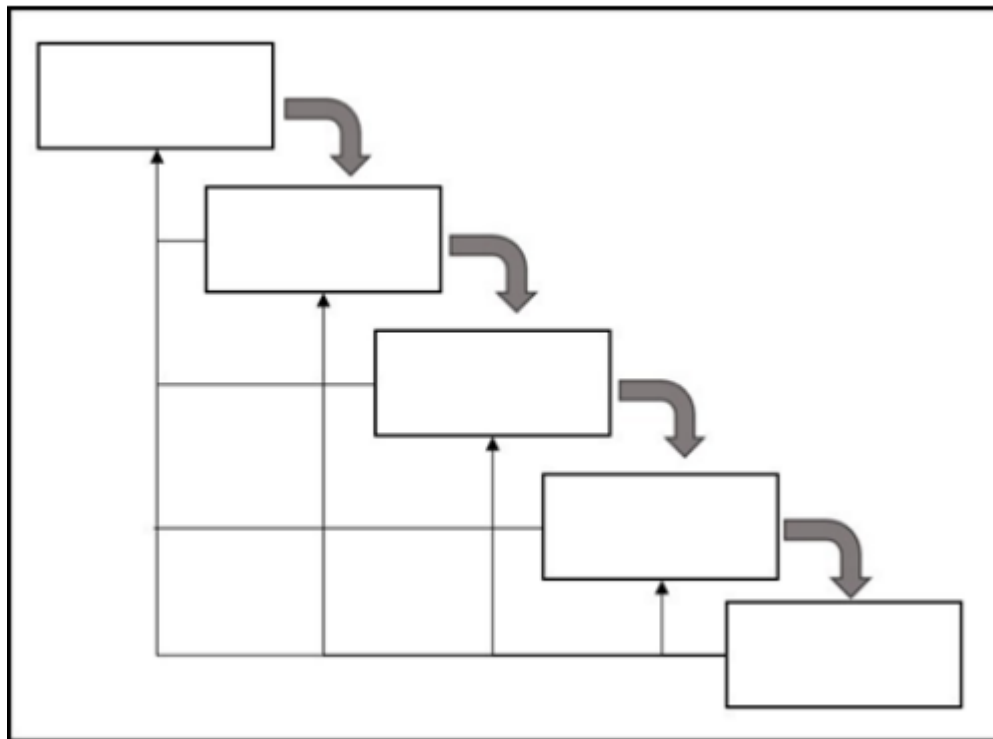
Verdadero.

Falso.

Ninguna de las opciones.

Justificación: La crisis del software surgió por los problemas que tenían las empresas para cumplir plazos y mantener la calidad en sus programas.

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



Gestión de proyecto, comunicación, planificación, diseño-modelado, construcción-implementación.

Comunicación, planificación, diseño-modelado, construcción-implementación, entrega-implantación.

Gestión de la configuración, comunicación, planificación, diseño-modelado, entrega-implantación

Justificación: Estas son las fases del modelo en cascada, que sigue un orden paso a paso desde la comunicación con el cliente hasta la entrega final.

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: Esa etapa no forma parte del modelo en espiral de Boehm, que se centra en planificación, análisis de riesgos y evaluación del cliente.

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: Un buen sistema de información parte de un proceso organizado, donde cada fase produce resultados que guían a la siguiente.

3. CONCLUSIONES

1. La ingeniería del software nació para solucionar los problemas de calidad y organización que había al crear programas. Su origen se relaciona con Margaret Hamilton y la Conferencia de la OTAN de 1968, donde se propuso tratar el desarrollo como una disciplina formal.
2. Un buen software debe ser fácil de mantener, confiable y sencillo de usar, ya que eso asegura su eficacia y aceptación.
3. El software no se fabrica, se desarrolla. No se desgasta con el uso, pero puede quedar obsoleto si no se actualiza o mantiene.

4. RECOMENDACIONES

1. Usar modelos como Cascada, Espiral o Ágil según el proyecto ayuda a mantener orden y calidad en cada fase del desarrollo.
2. Es mejor entregar un software bien hecho y confiable que algo rápido pero lleno de errores. La calidad siempre se nota a largo plazo.

3. Trabajar de la mano con el usuario ayuda a crear un producto que realmente cumpla con sus necesidades y expectativas.

5. REFERENCIAS

Pezzè, M. (2025). *A 2030 roadmap for software engineering*. *ACM Transactions on Software Engineering and Methodology*, 34(4), 1–25. <https://doi.org/10.1145/3731559>

Betz, S. (2025). *The role of software engineers in an AI era*. *ACM Transactions on Software Engineering and Methodology*, 34(3), 1–19. <https://doi.org/10.1145/3715112>

Ruiz, J. A., & Lucio, X. I. (2025). *Fundamentos de ingeniería de software: Un enfoque práctico*. Universidad de las Fuerzas Armadas-ESPE. ISBN: 978-9942-652-26-3