## 1. DATOS INFORMATIVOS

Carrera: Tecnologías de la Información

Asignatura: Metodologías de software

Tema del taller: Autoevaluación

Docente: Jenny A.Ruiz.R

Integrantes: Sebastian Lugmaña, Steeven Quishpi y Andoly Saguano

Fecha: 13/10/2025 Paralelo: 29022

## 2. DESARROLLO

Durante la autoevaluación del capítulo 1 se abordaron varias preguntas clave relacionadas con los fundamentos de la ingeniería de software, tomando como referencias principales a Pressman (2010) y Sommerville (2011). Las preguntas abarcaron temas desde la historia y definición del software, pasando por los atributos de un buen software, hasta modelos de procesos de desarrollo.

Se describen a continuación algunas preguntas y respuestas destacadas:

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

Margaret Hamilton.

Margaret Sanger.

Margaret Atwood.

Justificación :

Margaret Hamilton fue quien popularizó el término "ingeniería del software" durante su trabajo en el programa Apolo de la NASA en los años 60 (Sommerville, 2011).

**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 está formado por procedimientos, programas, datos relacionados, reglas y documentación que posibilitan que un sistema informático realice una función determinada. (Pressman, 2010)

**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 :

La primera vez que se empleó oficialmente el término "ingeniería de software" fue en la Conferencia OTAN de 1968, en la que se debatió sobre la llamada "crisis del software" y la importancia de aplicar principios de ingeniería al desarrollo de programas.

**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 métodos de razonamiento simbólico y algoritmos que no son numéricos para solucionar problemas complejos, emulando así la inteligencia humana.

**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 confiable, eficiente, usable y mantenible, cumpliendo con la funcionalidad y el rendimiento que el usuario requiere

**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 :

El software no se fabrica, sino que se desarrolla mediante procesos intelectuales y creativos, lo cual lo diferencia de los productos físicos tradicionales (Pressman, 2010).

**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.

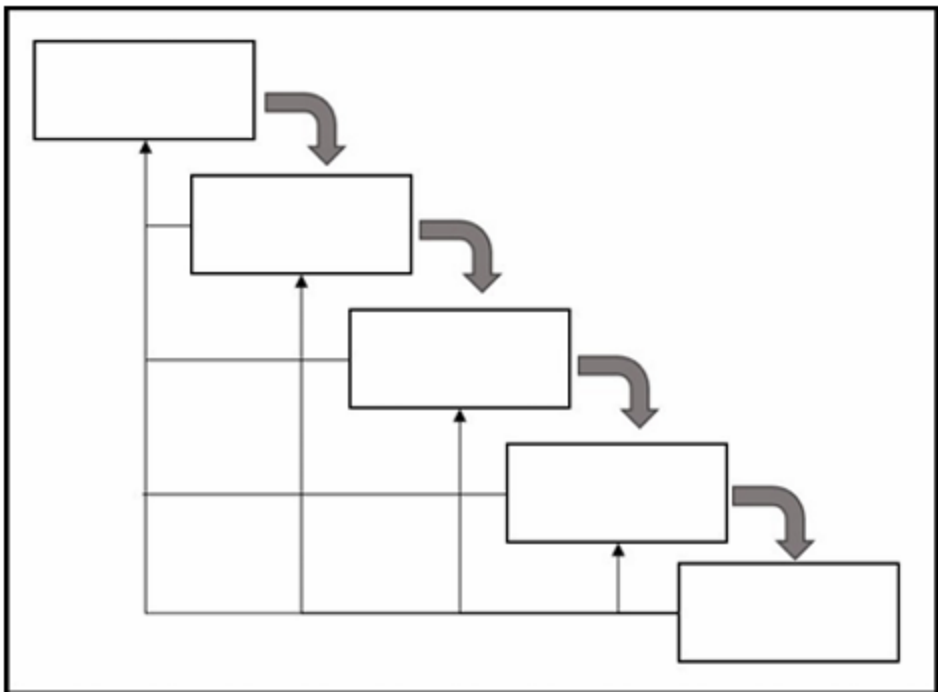
Falso.

Ninguna de las opciones.

Justificación :

La crisis del software describe los problemas de calidad, costos y retrasos en los proyectos de desarrollo de software durante los años 60 y 70, debido a la falta de métodos estructurados (Sommerville, 2011).

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



**Opciones:**

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 :

Boehm incluye las fases de planificación, análisis de riesgos, desarrollo y evaluación del cliente, pero no incluye la definición de un paradigma de desarrollo, ya que este es un concepto teórico, no una etapa del proceso (Boehm, 1988).

**9. El modelo de proceso de software en espiral propuesto por Boehm 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.

El modelo espiral, propuesto por Barry Boehm (1988), combina la naturaleza iterativa del desarrollo de prototipos con el enfoque sistemático del modelo en cascada

Justificación :

(Boehm, 1988, A Spiral Model of Software Development and Enhancement, IEEE Computer).

**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.

Para crear un sistema informático eficiente, es necesario iniciar con la implementación de un proceso de desarrollo estructurado también llamado modelo de proceso de software que fraccione el trabajo en etapas claramente definidas (análisis, diseño, implementación, pruebas y mantenimiento).

Justificación :

Sommerville, I. (2011). *Ingeniería del software* (9.ª ed.). Pearson Educación.

## 3. CONCLUSIONES

- La historia y evolución de la ingeniería del software están marcadas por la necesidad de superar problemas de calidad y gestión que surgieron con el crecimiento de los proyectos en los años 60.
- Las metodologías y modelos de desarrollo, como el modelo en cascada y el modelo espiral, buscan ordenar y estructurar el trabajo para obtener software de calidad.
- Los atributos de buen software incluyen funcionalidad, confiabilidad, eficiencia, mantenibilidad y usabilidad.
- Comprender los tipos de software y sus características permite diseñar mejores soluciones para diferentes contextos y necesidades.

## 4. RECOMENDACIONES

- Continuar estudiando modelos de proceso y metodologías ágiles para mejorar la adaptabilidad y eficiencia en el desarrollo.
- Promover la participación activa del cliente o usuario final durante todo el desarrollo para asegurar que el producto cumpla con sus expectativas.
- Incluir prácticas de documentación rigurosa que faciliten el mantenimiento y evolución futura del software.
- Realizar autoevaluaciones frecuentes para reforzar los conceptos y detectar áreas de mejora.

## 5. REFERENCIAS


Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5), 61–72. https://doi.org/10.1109/2.59

Encyclopaedia Britannica. (2024). Margaret Hamilton – American Computer Scientist. https://www.britannica.com/biography/Margaret-Hamilton-American-computer-scientist

IEEE Computer Society. (2020). What to Know About the Scientist Who Invented the Term "Software Engineering". https://www.computer.org/publications/tech-news/events/what-to-know-about-the-scientist-who-invented-the-term-software-engineering

Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). McGraw-Hill.

Sommerville, I. (2011). *Ingeniería del Software* (9ª ed.). Pearson Educación.

Velneo. (2023, July 12). 10 principales retos del desarrollo de software. https://velneo.com/blog/10-principales-retos-del-desarrollo-de-software/

Canelo, M. M. (2025, September 1). ¿Qué son los paradigmas de programación? Profile Software Services. https://profile.es/blog/que-son-los-paradigmas-de-programacion/