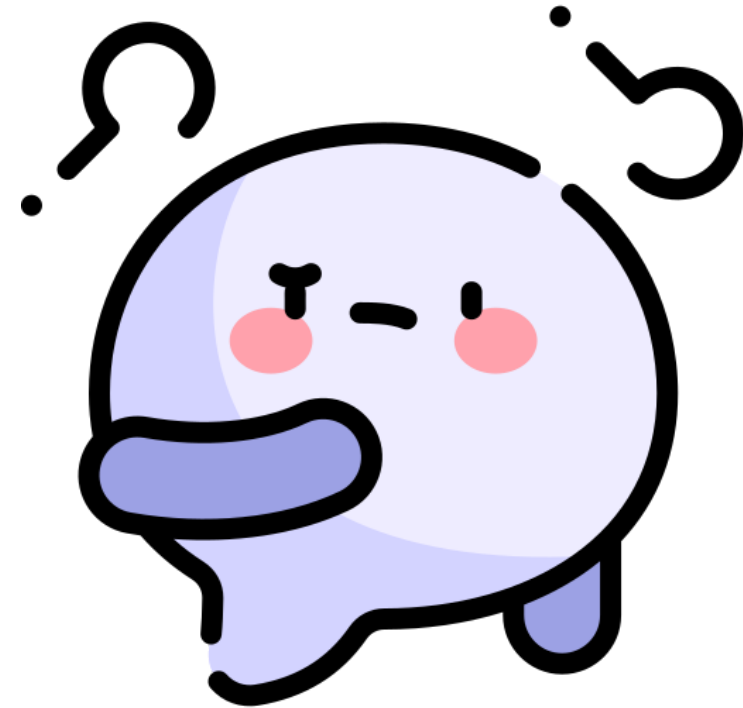


Introducción y aplicación del SWEBOK.

¿Qué es el SWEBOK?

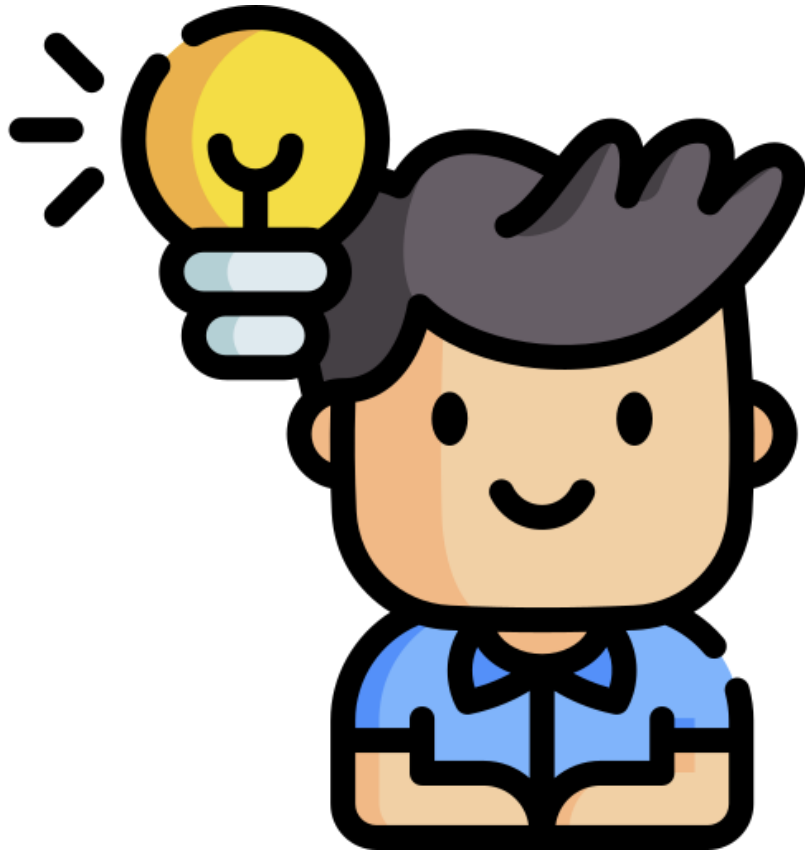
- **Son las siglas:**
- **Software Engineering Body of Knowledge.**
- **La cual es una guía actualizada sobre el conocimiento aceptado y consensuado respecto a la teoría y la práctica de la Ingeniería de Software**



¿Cuáles son sus propósitos?

- Promover una visión consistente de la Ing. de Software en todo el mundo.
- Especificar el alcance y aclarar el lugar de la Ing. de Software respecto a otras disciplinas
- Caracterizar los contenidos.
- Proporcionar acceso temático al cuerpo de conocimientos de la materia
- Proporcionar una base para el desarrollo curricular y para la certificación





El SWEBOK ha ido sufriendo diversas revisiones a lo largo de su historia.

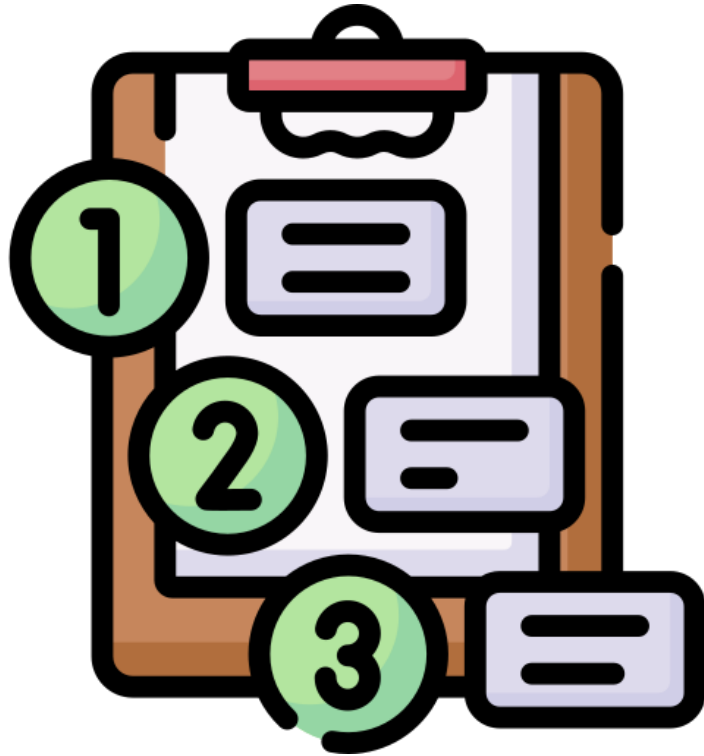
Así, en 2004 se publicaba la primera versión, el SWEBOK 2004 (estándar ISO/IEC TR 19759:2005).

Después, en 2024 se publicó la revisión más reciente, el SWEBOK V4.0 (estándar ISO/IEC TR 19759:2015).

A nivel de contenidos, el SWEBOK V3.0 organiza el ámbito en 18 áreas de conocimiento (Knowledge Areas o KAs).

1. **Software Requirements – Requisitos de Software**
2. **Software Architecture – Arquitectura de Software**
3. **Software Design – Diseño de Software**
4. **Software Construction – Construcción de Software**
5. **Software Testing – Pruebas de Software**
6. **Software Engineering Operations – Operaciones de Ingeniería de Software**
7. **Software Maintenance – Mantenimiento de Software**
8. **Software Configuration Management – Gestión de la Configuración de Software**
9. **Software Engineering Management – Gestión de la Ingeniería de Software**





10. **Software Engineering Process – Proceso de Ingeniería de Software**
11. **Software Engineering Models and Methods – Modelos y Métodos de Ingeniería de Software**
12. **Software Quality – Calidad de Software**
13. **Software Security – Seguridad de Software**
14. **Professional Practice of Software Engineering – Práctica Profesional de Ingeniería de Software**
15. **Software Engineering Economics – Economía de la Ingeniería de Software**
16. **Computing Fundamentals – Fundamentos de Computación**
17. **Mathematical Fundamentals – Fundamentos Matemáticos**
18. **Engineering Fundamentals – Fundamentos de Ingeniería**

Breve descripción de las 18 áreas del conocimiento



Categoría

1. Requerimientos del software



2. Arquitectura del Software



Definición

1. **Propiedades o capacidades que el software debe tener para satisfacer necesidades de usuarios y sistemas.**
2. **Estructura fundamental de un sistema, definiendo sus componentes clave y cómo interactúan entre sí.**

Categoría

3. Diseño del Software



4. Construcción del Software



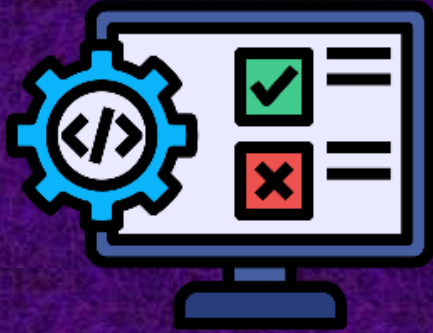
Definición

3. Proceso de planear la arquitectura, componentes, interfaces y características de un sistema, así como el resultado de dicho proceso.

4. Creación del código ejecutable a partir de diseños, utilizando lenguajes de programación y herramientas.

Categoría

5. Pruebas del Software



6. Operaciones de Ingeniería de Software



Definición

5. Evaluación del software para detectar errores y verificar que cumple con los requerimientos especificados.

6. Actividades relacionadas con el despliegue, monitoreo y soporte del software en entornos reales.

Categoría

7. Mantenimiento del Software



8. Gestión de Configuración del Software



Definición

7. Modificaciones posteriores al lanzamiento para corregir fallos, mejorar rendimiento o adaptarse a nuevos entornos.

8. Control de cambios en el software durante su ciclo de vida, incluyendo versiones y actualizaciones.

Categoría

9. Gestión de la Ingeniería de Software



10. Proceso de Ingeniería de Software



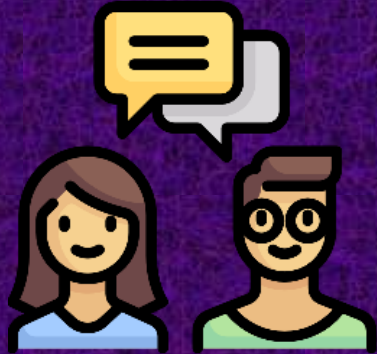
Definición

9. Planificación, coordinación y supervisión de recursos, plazos y equipos en proyectos de software.

10. Conjunto de actividades, métodos y prácticas utilizadas para desarrollar software de manera sistemática.

Categoría

11. Modelos y Métodos de Ingeniería de Software



12. Calidad del Software



Definición

11. Técnicas y enfoques (como ágil, cascada o DevOps) para guiar el desarrollo y gestión de proyectos.

12. Conjunto de características que aseguran que el software cumple con estándares y expectativas definidas.

Categoría

13. Seguridad del Software



14. Práctica Profesional de la Ingeniería de Software



Definición

13. Protección del software contra amenazas, vulnerabilidades y accesos no autorizados.

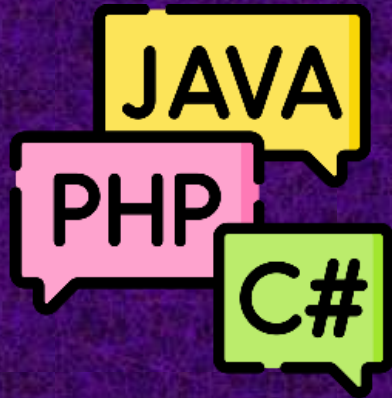
14. Normas éticas, responsabilidades y competencias requeridas para ejercer como ingeniero de software.

Categoría

15. Economía de la Ingeniería de Software



16. Fundamentos de Computación



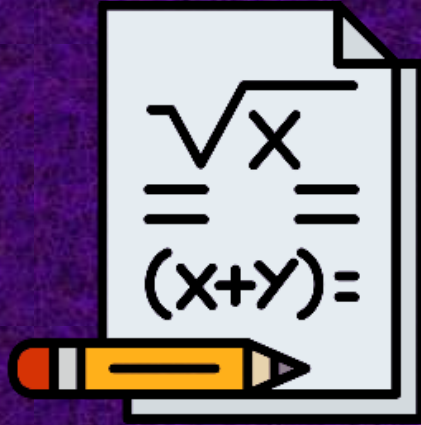
Definición

15. Análisis de costos, beneficios y viabilidad financiera en la toma de decisiones de proyectos de software.

16. Bases teóricas y prácticas de la informática, como algoritmos, estructuras de datos y sistemas operativos.

Categoría

17. Fundamentos Matemáticos



18. Fundamentos de Ingeniería



Definición

17. Principios matemáticos aplicados al software, como lógica, estadística y teoría de grafos.

18. Principios generales de ingeniería (como gestión de riesgos o toma de decisiones) aplicados al desarrollo de software.

¿Por qué SWEBOK es útil para los desarrolladores de software?

- Los desarrolladores de software pueden utilizar SWEBOK para evaluar sus habilidades y carencias en diversos temas de ingeniería de software y encontrar recursos para aprender más sobre los principios y conocimientos de la misma.
- SWEBOK también puede ayudar a los desarrolladores a gestionar el progreso de sus competencias al ofrecer una estructura y un lenguaje común para las ideas y los métodos del campo.

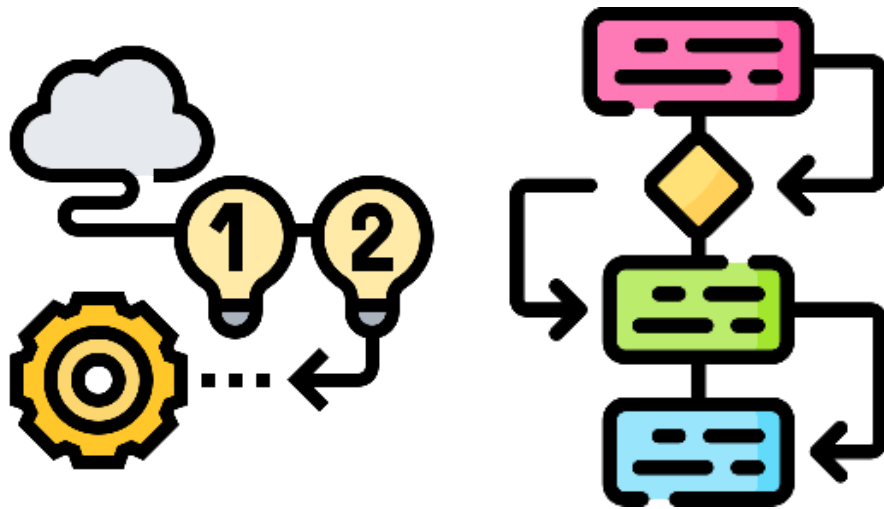


¿Cómo pueden los desarrolladores de software utilizar SWEBOK?



- **Comprobar sus propios conocimientos y experiencia en cada AC (Área de Conocimiento).**
- **Obtener más información de las referencias y enlaces de cada AC, así como de los temas relacionados y emergentes.**

¿Cómo pueden los desarrolladores de software utilizar SWEBOK?



- Como una guía para comprender y utilizar términos, conceptos, mejores prácticas y estándares de ingeniería de software.
- Para planificar su desarrollo profesional, estableciendo objetivos de competencia basados en los temas.

Fuentes de consulta:

- **easymaint.net. (s.f.). *Explorando el Mantenimiento de Software: Garantizando la Salud y Eficiencia de tus Aplicaciones Digitales*. Obtenido de [https://www.easymaint.net/faq/que-es-el-mantenimiento-de-software.html#:~:text=Definici%C3%B3n%20de%20Mantenimiento%20de%20Software:%20El%20mantenimiento,correctivo\)%2C%20la%20implementaci%C3%B3n%20de%20nuevas%20caracter%C3%ADsticas%20\(](https://www.easymaint.net/faq/que-es-el-mantenimiento-de-software.html#:~:text=Definici%C3%B3n%20de%20Mantenimiento%20de%20Software:%20El%20mantenimiento,correctivo)%2C%20la%20implementaci%C3%B3n%20de%20nuevas%20caracter%C3%ADsticas%20()**
- **Hironori Washizaki, M.-I. S.-S. (s.f.). *An Overview of the SWEBOK Guide*. Obtenido de https://sebokwiki.org/wiki/An_Overview_of_the_SWEBOK_Guide#Introduction**
- **Society, I. C. (s.f.). *Software Engineering Body of Knowledge (SWEBOK)*. Obtenido de <https://www.computer.org/education/bodies-of-knowledge/software-engineering>**
- **UNAM, D. G. (s.f.). *Software Engineering Body of Knowledge (SWEBOK)*. Obtenido de <https://www.red-tic.unam.mx/content/software-engineering-body-knowledge-swebok>**
- **Univeritat Oberta de Catalunya por Viladrosa, R. C. (s.f.). *¿Qué debes saber para desarrollar software de forma profesional?* Obtenido de <https://blogs.uoc.edu/informatica/es/que-debes-saber-para-desarrollar-software-de-forma-profesional/>**