

El Ciclo de Vida del Software (SDLC, por sus siglas en inglés) constituye un marco fundamental para comprender las etapas que intervienen en la creación de sistemas informáticos: desde la conceptualización y el diseño, hasta el desarrollo, las pruebas, el despliegue y el mantenimiento.

Conocer y aplicar adecuadamente el SDLC resulta clave para asegurar la calidad, la seguridad y la eficiencia de los proyectos de software, independientemente de si se adoptan metodologías clásicas o enfoques ágiles.

Esta guía presenta un desglose de las fases principales del ciclo, así como las herramientas más relevantes en cada una de ellas. A lo largo del estudio se enfatiza un principio central: la tecnología, en sí misma, no es la causa de los problemas en los sistemas; más bien, las dificultades suelen originarse en un diseño inadecuado o deficiente.

A- Conceptos: Responde de forma conceptual y sintética el siguiente cuestionario

1. ¿Qué es el Ciclo de Vida del Software y por qué es importante conocerlo? El
2. ¿Cuál es la diferencia fundamental entre la fase de Análisis y la fase de Diseño en el SDLC
3. Según el material, ¿dónde radica el principal problema de las aplicaciones que fallan o son ineficientes
4. Menciona y describe brevemente dos tipos de pruebas realizadas en la fase de Pruebas del SDLC.
5. ¿Qué implica la fase de Despliegue y por qué se considera un proceso complejo?
6. ¿Cuál es el propósito principal de la fase de Mantenimiento?
7. Compara brevemente las "historias de usuario" y los "casos de uso" como herramientas de análisis.
8. ¿Qué son los mockups y cuál es su utilidad en la fase de Diseño?
9. Menciona al menos tres herramientas o conceptos clave en la fase de Desarrollo del software.
10. Explica por qué la seguridad es un tema importante a considerar en el Ciclo de Vida del Software y en qué etapas puede abordarse.

B- Análisis y reflexión: desarrollar los siguientes temas propuestos basados en el texto e investigación sobre las etapas y herramientas del SDLC.

1. Analice críticamente la afirmación sobre que *la tecnología no es la culpable de los problemas en el desarrollo de software, sino el diseño*. ¿Está de acuerdo? Justifique su respuesta con ejemplos y argumentos basados en el texto.
2. Compare y contraste las herramientas de análisis para metodologías clásicas (casos de uso, UML) y ágiles (historias de usuario, scrum), destacando sus enfoques y beneficios para el levantamiento de requerimientos.
3. Explique la importancia de las pruebas de usabilidad, seguridad y calidad dentro del Ciclo de Vida del Software, y cómo una omisión o deficiencia en cualquiera de estas puede afectar el éxito de un proyecto.
4. Describa los principales desafíos y consideraciones al seleccionar herramientas y plataformas para la fase de despliegue de software, haciendo énfasis en la evolución de los servidores locales a la infraestructura en la nube y los servicios de despliegue móvil.
5. Reflexione sobre la relevancia de la capacitación y especialización profesional en las diferentes etapas del Ciclo de Vida del Software, considerando las oportunidades tanto para trabajar en una organización como de manera independiente.