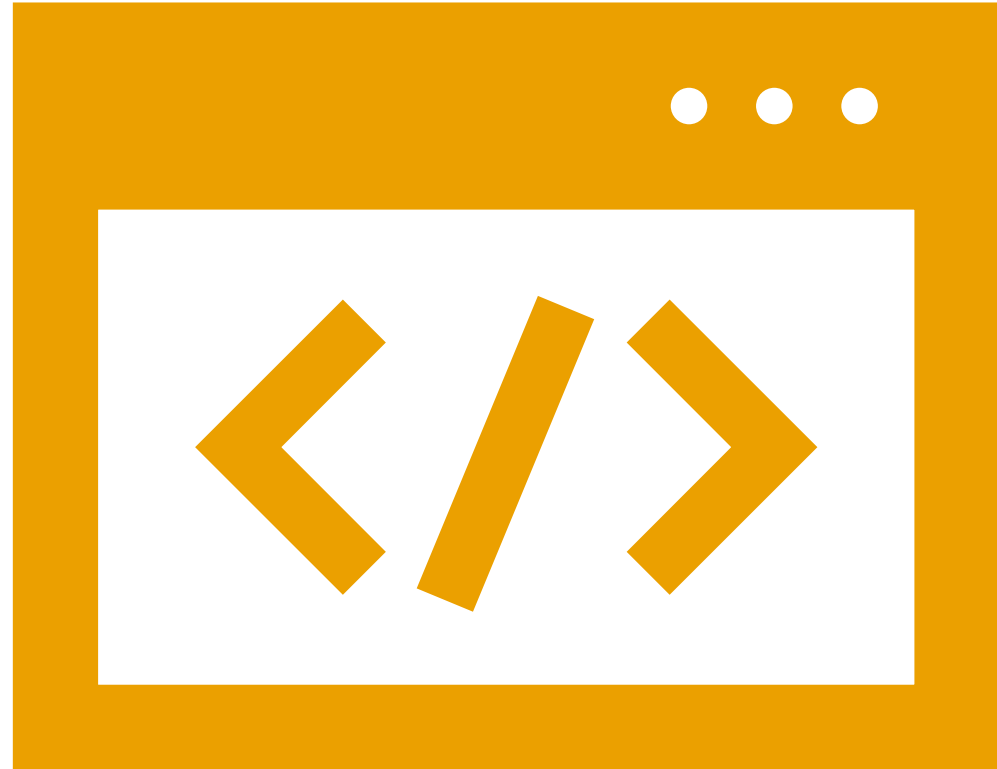

TEMAS SELECTOS DE INGENIERÍA DE SOFTWARE

*El ciclo de vida y el plan de trabajo
con base en el Proceso Unificado*

¿Creen que el desarrollo de software termina cuando lanzamos el producto? ¿Por qué?



¿Qué es el ciclo de vida del software?

"El ciclo de vida del desarrollo de software (Software Development Life Cycle - SDLC) es un conjunto de procesos, actividades y tareas aplicados durante la adquisición, suministro, desarrollo, operación, mantenimiento y disposición del software, diseñado para lograr la satisfacción del cliente mediante la planificación, control y mejora continua de esas actividades."

- Definición oficial IEEE/ISO/IEC 12207 (estándar internacional)

¿Qué es el ciclo de vida del software?

El ciclo de vida del software es un marco conceptual que define las fases, desde la concepción y especificación de requisitos hasta el mantenimiento, donde el software es desarrollado y evolucionado mediante modelos formales para garantizar calidad y rendimiento.

- Roger S. Pressman

¿Por qué es Importante el SDLC?

Beneficios Clave:

✓ Control de calidad - Reduce errores y defectos





✓ Gestión de riesgos - Identifica problemas temprano

✓ Control de costos - Evita sobrecostos por retrabajos

✓ Planificación efectiva - Define tiempos y recursos

✓ Satisfacción del cliente - Cumple expectativas

Fases Fundamentales del SDLC

1.  Planificación y Análisis de Requisitos
2.  Diseño del Sistema
3.  Implementación/Desarrollo
4.  Verificación y Pruebas
5.  Despliegue
6.  Mantenimiento y Evolución

¿En qué fase creen que suelen ocurrir más errores y por qué?

Opciones:

A) Análisis de requisitos

B) Diseño

C) Implementación

D) Pruebas

Costos de Corrección de Errores:

Requisitos: \$1

Diseño: \$5

Implementación: \$10

Pruebas: \$20

Producción: \$200

Conclusión: La detección temprana de errores es crítica

Ejemplo práctico

Caso: Aplicación de Banca Móvil

Planificación: Identificar funcionalidades (transferencias, consultas)

Diseño: Arquitectura segura, interfaz intuitiva

Implementación: Código, APIs, bases de datos

Pruebas: Seguridad, usabilidad, rendimiento

Despliegue: App stores, servidores

Mantenimiento: Actualizaciones de seguridad, nuevas funciones

Mitos vs. Realidades

MITO	REALIDAD
El software nunca falla	Requiere mantenimiento continuo
Solo programar es suficiente	60% del tiempo es análisis y diseño
Los cambios son fáciles	Los cambios tardíos son costosos
La documentación es opcional	Es esencial para mantenimiento

Evolución del SDLC

Tendencias Actuales:



De lineal a iterativo - Metodologías ágiles



DevOps y CI/CD - Integración continua



Automatización - Testing automático



Desarrollo continuo - Actualizaciones frecuentes

Pregunta de Reflexión Final



¿Cómo ha transformado la inteligencia artificial el ciclo de vida del software?

Consideren:

Generación automática de código

Testing inteligente

Análisis predictivo de errores

Personalización continua

Puntos Clave para Recordar

✨ El SDLC es más que programación

✨ La planificación temprana ahorra costos

✨ La calidad se construye en cada fase

✨ El mantenimiento es la fase más larga

✨ La adaptabilidad es clave en el mundo actual
