

## 1.4 SLDC, UML y ERP

**1.4.1 Desarrolla notas de clase relacionada con UML mencionando tu autoría con los siguientes puntos:**

- (a) *¿Quienes fueron sus creadores iniciales?*
- (b) *¿Qué es y para qué sirve?*
- (c) *¿Qué relación tiene con el SLDC?*
- (d) *Conclusiones.*

Notas de clase – UML (Lenguaje Unificado de Modelado)

Autoría: Kimi (K2) – 2026-01-05

a) Creadores iniciales

UML nació en 1994-1997 de la unificación de los métodos OOSE (Ivar Jacobson), OMT (James Rumbaugh) y Booch (Grady Booch). El grupo “Tres Amigos” presentó UML 1.0 al OMG en 1997, convirtiéndose en estándar abierto.

b) ¿Qué es y para qué sirve?

UML es un lenguaje gráfico estandarizado para visualizar, especificar, construir y documentar sistemas software.

Sirve para:

- Comunicar requisitos y diseño entre técnicos y negocio.
- Generar código esqueleto (CASE) y casos de prueba.
- Documentar arquitectura, bases de datos, procesos de negocio.

c) Relación con el SDLC

- Fases tempranas (Planificación/Análisis): diagramas de casos de uso y de actividades capturan requisitos funcionales.
- Diseño: clases, objetos, secuencia, componentes y despliegue definen arquitectura antes de codificar.
- Implementación: los diagramas sirven de “planos” para los desarrolladores y validación posterior.
- Mantenimiento: al cambiar el código se actualiza el modelo UML para preservar la trazabilidad.

d) Conclusiones

UML no es obligatorio, pero **reduce la brecha de comprensión** entre negocio y técnico, **acorta re-trabajos** y **aumenta la calidad** del software dentro de cualquier SDLC (cascada, ágil o DevOps). **Usar solo los diagramas que aporten valor es la práctica más común hoy.**

## **1.4.2 ¿Reduce la brecha de comprensión entre negocio y técnico? ¿Cuál es la relación con ERP?**

Sí: al expresar los procesos con símbolos gráficos compartidos (casos de uso, actividades, estados, etc.) UML deja de lado el "código" y habla el lenguaje de los analistas; eso **reduce la brecha** porque ambos lados ven el mismo modelo y pueden discutir sobre él sin ambigüedad.

Relación con un ERP:

### 1. Levantamiento de requisitos

- Casos de uso UML describen escenarios de compras, ventas, inventario, nómina, etc.
- Mapas de procesos (diagramas de actividad) muestran cómo fluyen los documentos entre módulos del ERP.

### 2. Configuración / parametrización

- Los objetos del modelo UML (clases "Artículo", "Cliente", "Orden") se traducen directamente en tablas o vistas maestras del ERP.
- Diagramas de secuencia definen qué servicios internos debe orquestar el ERP (validar stock, reservar, facturar).

### 3. Personalizaciones y extensiones

- Antes de crear un desarrollo ad-hoc (reporte, interfaz, web-service) se modela la lógica en UML; así se valida que la extensión no rompa el núcleo estándar del ERP.

### 4. Documentación de soporte

- Mantener los diagramas UML actualizados sirve como "manual técnico" para futuras auditorías o nuevos consultores del ERP.

En resumen: UML **aclara** qué necesita el negocio; el ERP **materializa** esa lógica en un paquete software. Los diagramas UML actúan de **interprete común** entre la empresa y el equipo de implantación, reduciendo la brecha y los costos de re-configuración.

## **1.4.3 Pero no has explicado que es un ERP**

ERP (Enterprise Resource Planning) es un **sistema de planificación de recursos empresariales**: software integrado que **centraliza y automatiza los procesos clave** de una organización (finanzas, compras, ventas, inventario, producción, RRHH, etc.) en **una única base de datos** y **una interfaz común** para todos los departamentos.