

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



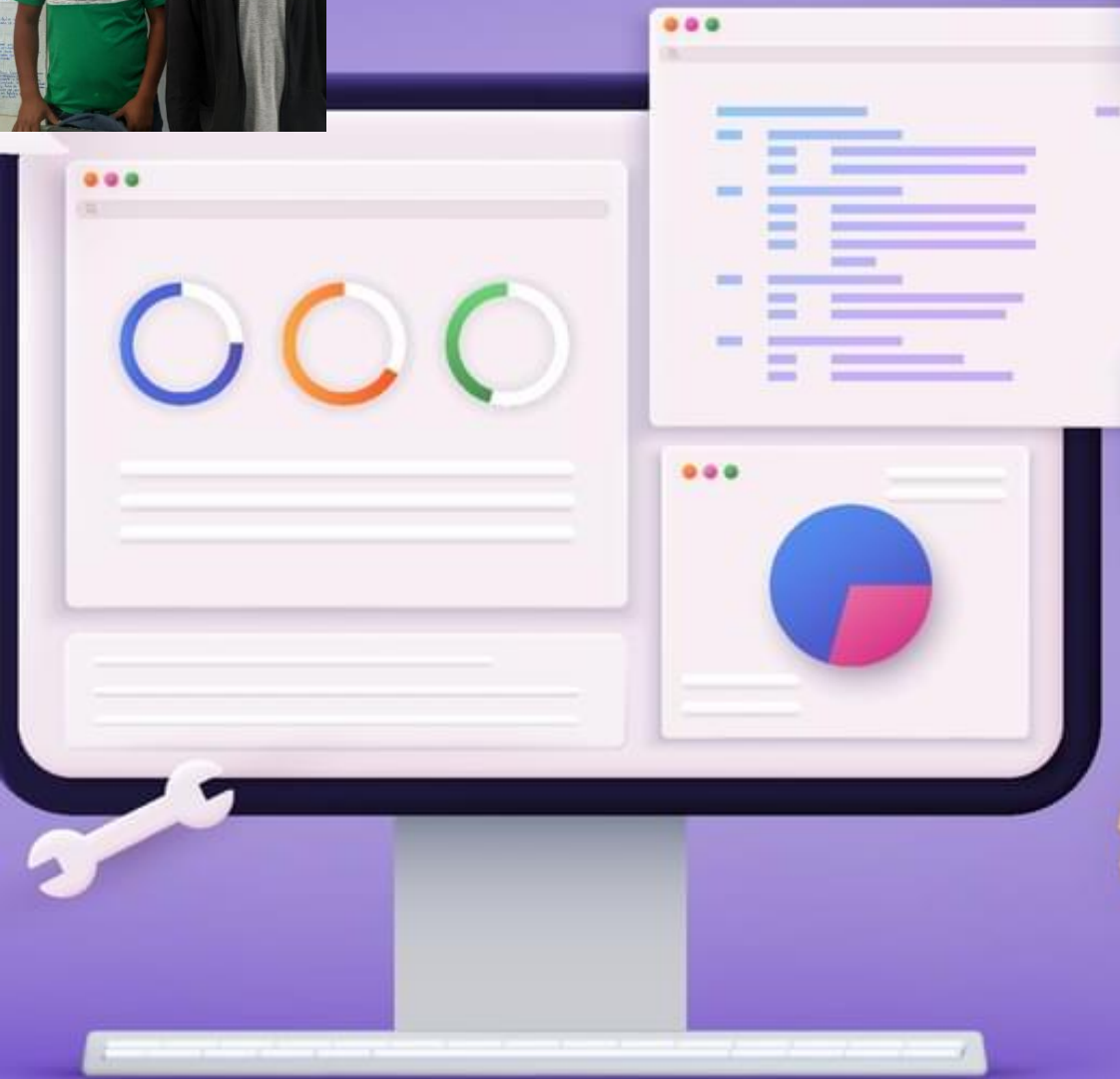
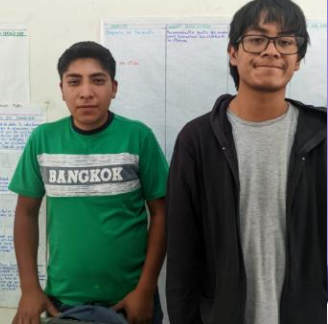
Diseño de sistemas de información

Fecha: 3 de abril de 2025

Alberto Alejandro Morales Caballero

Materia: Análisis y Modelado de Sistemas de información 4B





SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



Modelo de datos en UML

Fecha: 3 de abril de 2025

Alberto Alejandro Morales Caballero

Materia: Análisis y Modelado de Sistemas
de información



Introducción

- + El modelo de datos en UML tiene como finalidad representar la estructura y organización de los datos dentro de un sistema, facilitando la visualización de entidades, sus atributos y las relaciones entre ellas. Este modelo es esencial para diseñar bases de datos o estructuras de información, asegurando coherencia y eficiencia. Utiliza símbolos como entidades (rectángulos), atributos (elipses conectadas a entidades) y relaciones (líneas con cardinalidades), que permiten mapear cómo interactúan los elementos en el sistema.



Modelo de Datos en UML



Entidades

Representan objetos, conceptos o elementos clave del sistema que almacenan información.

- Ejemplos:
 - En un sistema de ventas: Cliente, Producto, Pedido.
 - En un hospital: Paciente, Médico, Cita.
 - En una escuela: Un profesor, una clase, alumnos.

Atributos

- Características o propiedades específicas de una entidad.
- Ejemplos:
 - Cliente: ID_Cliente (PK), Nombre, Correo.
 - Producto: Código (PK), Precio, Stock.

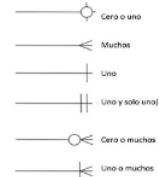
- Tipos:
 - Clave primaria (PK): Subrayado o etiqueta «PK» → Identificador único (ej: ID_Cliente).
 - Clave foránea (FK): Etiqueta «FK» → Vincula entidades (ej: ID_Pedido en DetallePedido).

Relaciones

- Definen cómo interactúan las entidades entre sí.
- Ejemplos:
 - Cliente "realiza" Pedido (uno a muchos).
 - Empleado "trabaja en" Departamento (muchos a uno).

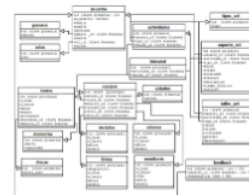
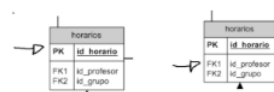
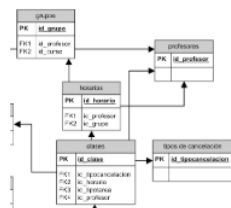
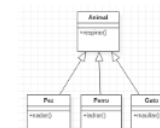
- Cardinalidades:
- 1: Relación uno a uno (ej: Persona — Pasaporte).
 - 1..*: Uno a muchos (ej: Cliente — Pedido).
 - 0..1: Opcional (ej: Empleado puede o no tener AutoEmpresa).
 - *: Muchos (ej: Estudiante — Curso).

Cardinalidad



Herencia/Generalización

- Permite que entidades hereden atributos y relaciones de una entidad general.
- Ejemplo:
 - Empleado y Cliente heredan de Persona (atributos: Nombre, Teléfono).



Conclusión

- + En resumen, el modelo de datos en UML es una herramienta fundamental para plasmar la arquitectura de la información, garantizando claridad en el diseño de bases de datos. Su uso de símbolos estandarizados facilita la comunicación entre equipos técnicos y no técnicos, promoviendo un desarrollo sistemático y evitando ambigüedades en la implementación.





SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



Modelo de clases en UML

Fecha: 3 de abril de 2025

Alberto Alejandro Morales Caballero

Materia: Análisis y Modelado de Sistemas de información



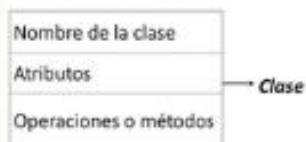
Introducción

- + El modelo de clases en UML busca definir la estructura estática de un sistema mediante la representación de clases, sus atributos, métodos y las relaciones entre ellas (como herencia, asociación o composición). Este diagrama es clave para planificar la arquitectura de software, ya que organiza los componentes principales del sistema y sus interacciones. Los símbolos incluyen clases (rectángulos divididos en nombre, atributos y métodos), flechas para herencia (triángulo hueco) y líneas con rótulos para asociaciones.



Modelo de Clases en UML

Clases



Concepto

Representan entidades con estructura (atributos) y comportamiento (métodos) en el sistema. Son la base del diseño orientado a objetos.

- En un sistema bancario: CuentaBancaria (atributos: saldo, númeroCuenta; métodos: depositar(), retirar()).
- En una red social: Usuario (atributos: nombreUsuario, contraseña; métodos: publicar(), seguir()).

Atributos



Propiedad o característica de una clase que describe su estado. Puede ser de tipos primitivos (ej: int) o referencias a otras clases.

Tipos

- Público (+): Accesible desde cualquier clase.
- Privado (-): Solo accesible dentro de su clase.
- Protegido (#): Accesible en la clase y sus subclases.



Método



Acción o operación que una clase puede realizar. Define el comportamiento de los objetos.

- De acceso (Getters/Setters): Controlan el acceso a atributos privados. Ejemplo: +getSaldo(): double.
- De negocio: Implementan lógica específica (ej: +calcularIntereses()).
- Estáticos: Pertenecen a la clase, no a instancias (ej: +generarID()).



1 no mas de uno
 0..1 cero o uno
 0..* cero o muchos
 1..* uno o mucho
 * muchos

Multiplicidad
 (Numero de elementos
 relacionados)

Modelo de Clases en UML

Asociación

Concepto

Conexión que indica una relación estructural entre clases, donde una clase utiliza o interactúa con otra.

———— Asociación

- Unidireccional: Flecha indica la dirección (ej: Cliente → realiza → Pedido).
- Bidireccional: Sin flecha (ej: Profesor ↔ enseña ↔ Estudiante).

Agregación

Relación "tiene-un" débil: un objeto contiene otros, pero estos existen independientemente.

◊ — Agregación

- Ejemplo:
- Universidad ◊ — Departamento (la universidad puede cerrar, pero los departamentos siguen existiendo).

Composición

Relación "tiene-un" fuerte: el objeto contenedor gestiona el ciclo de vida de las partes.

◆ — Composición

- Ejemplo:
- Automóvil ◆ — Motor (si se destruye el automóvil, el motor también).

Herencia/Generalización

Una clase (subclase) hereda atributos y métodos de otra (superclase).

————> Herencia

- Ejemplo:
- Gato ◊ — Animal (hereda nombre y método comer()).



Modelo de Clases en UML

1 no mas de uno
 0..1 cero o uno
 0..* cero o muchos
 1..* uno o mucho
 * muchos

Multiplicidad
(Numero de elementos relacionados)

Dependencia

Concepto

Indica que una clase usa temporalmente otra, sin una relación estructural.

-----> Dependencia

- Ejemplo:
- Factura ----> PDFGenerator (la factura necesita generar un PDF, pero no lo almacena).

Interfaz

Contrato que define métodos que una clase debe implementar. No contiene atributos ni implementaciones.

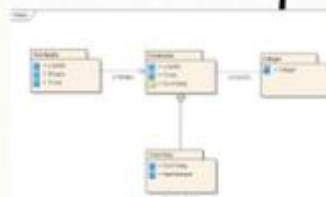
- Ejemplo:
- <<interface>> Notificable con método enviarNotificacion().
- CorreoElectronico y SMS implementan esta interfaz.



Paquete

Mecanismo para agrupar clases relacionadas, facilitando la organización en sistemas grandes. Símbolo: Icono de carpeta.

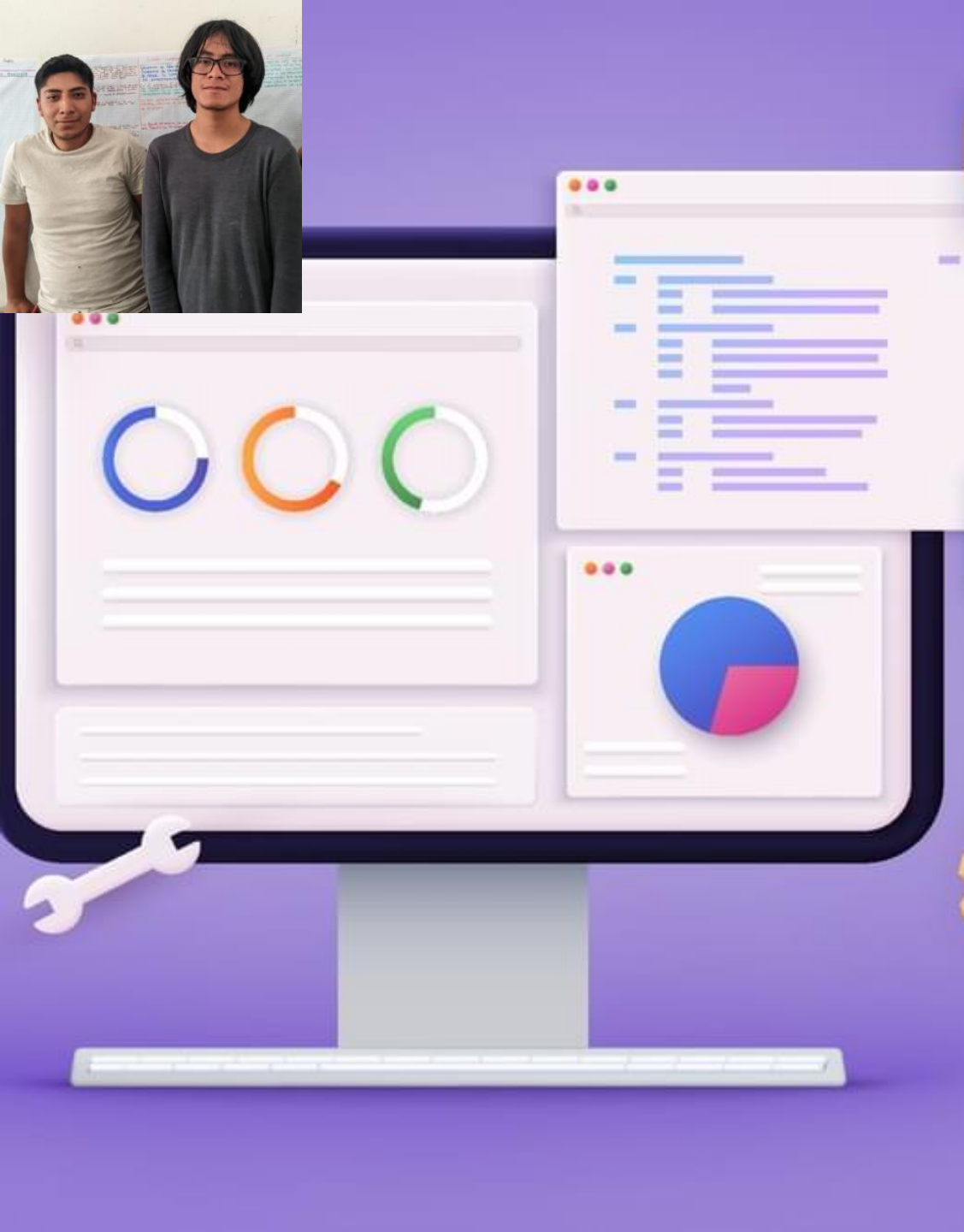
- Ejemplo:
- Paquete RecursosHumanos agrupa Empleado, Departamento, Nómina.



Conclusión

- + El modelo de clases destaca por su capacidad para esquematizar la estructura lógica de un sistema, sirviendo como base para la codificación. Su precisión en la definición de relaciones y responsabilidades entre clases optimiza el desarrollo y mantenimiento del software, asegurando escalabilidad y modularidad.





SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



Diagramas de secuencia en UML

Fecha: 1 de Abril del 2025

Alberto Alejandro Morales Caballero

Materia: Análisis y Modelado de Sistemas
de información

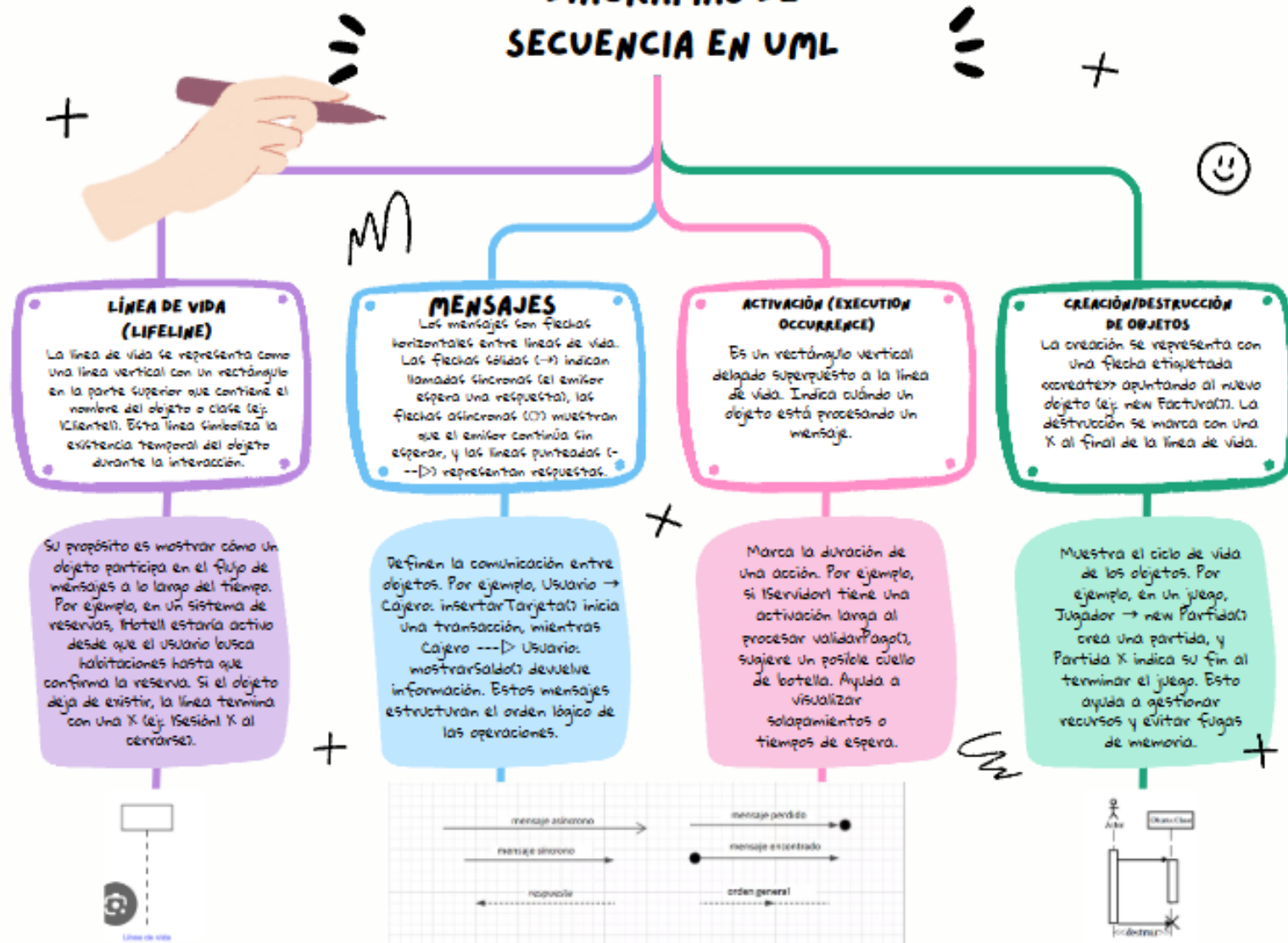


Introducción

- + Los diagramas de secuencia en UML tienen como objetivo modelar la interacción dinámica entre objetos o componentes de un sistema a lo largo del tiempo. Son ideales para visualizar flujos de procesos, como llamadas a métodos o respuestas entre entidades. Utilizan símbolos como líneas de vida (rectángulos verticales que representan objetos), mensajes (flechas etiquetadas) y barras de activación (rectángulos delgados que indican la ejecución de un método).



DIAGRAMAS DE SECUENCIA EN UML



Conclusión

- + Estos diagramas son esenciales para entender el comportamiento temporal de un sistema, identificando posibles cuellos de botella o errores en la lógica de interacción. Su enfoque en la secuenciación y sincronización de eventos los convierte en una herramienta invaluable para el diseño y depuración de sistemas complejos.





SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



Modelo de interfaces en UML

Fecha: 1 de Abril del 2025

Alberto Alejandro Morales Caballero

Materia: Análisis y Modelado de Sistemas
de información



Introducción

- + El modelo de interfaces en UML se centra en definir cómo los componentes de un sistema interactúan entre sí, especificando contratos o protocolos de comunicación. Es crucial para garantizar la interoperabilidad entre módulos, especialmente en sistemas distribuidos. Los símbolos incluyen interfaces (representadas con círculos o rectángulos con el estereotipo <<interface>>) y conexiones de realización (líneas punteadas con flechas triangulares).





Conclusión

- + Al establecer contratos claros entre componentes, este modelo asegura que las partes de un sistema funcionen de manera cohesionada, independientemente de su implementación interna. Su uso promueve la modularidad y la reutilización de código, aspectos clave en el desarrollo de software robusto y mantenible.



Bibliografía

Fowler, M. (2003). UML distilado: Una guía breve del lenguaje estándar de modelado de objetos (3ª ed.). Addison-Wesley.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). Guía del usuario de Unified Modeling Language (2ª ed.). Addison-Wesley.

Larman, C. (2004). Aplicando UML y patrones: Introducción al análisis y diseño orientado a objetos y al desarrollo iterativo (3ª ed.). Prentice Hall.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Análisis y diseño de sistemas con UML (5ª ed.). Wiley.

