



Modelado Unificado **UML**

Profesor:
Francisco Jimenez Bonilla
Estudiantes:
Christian Fallas Mora
Thais Hernández Quesada
Jhonny Fernandez Solano

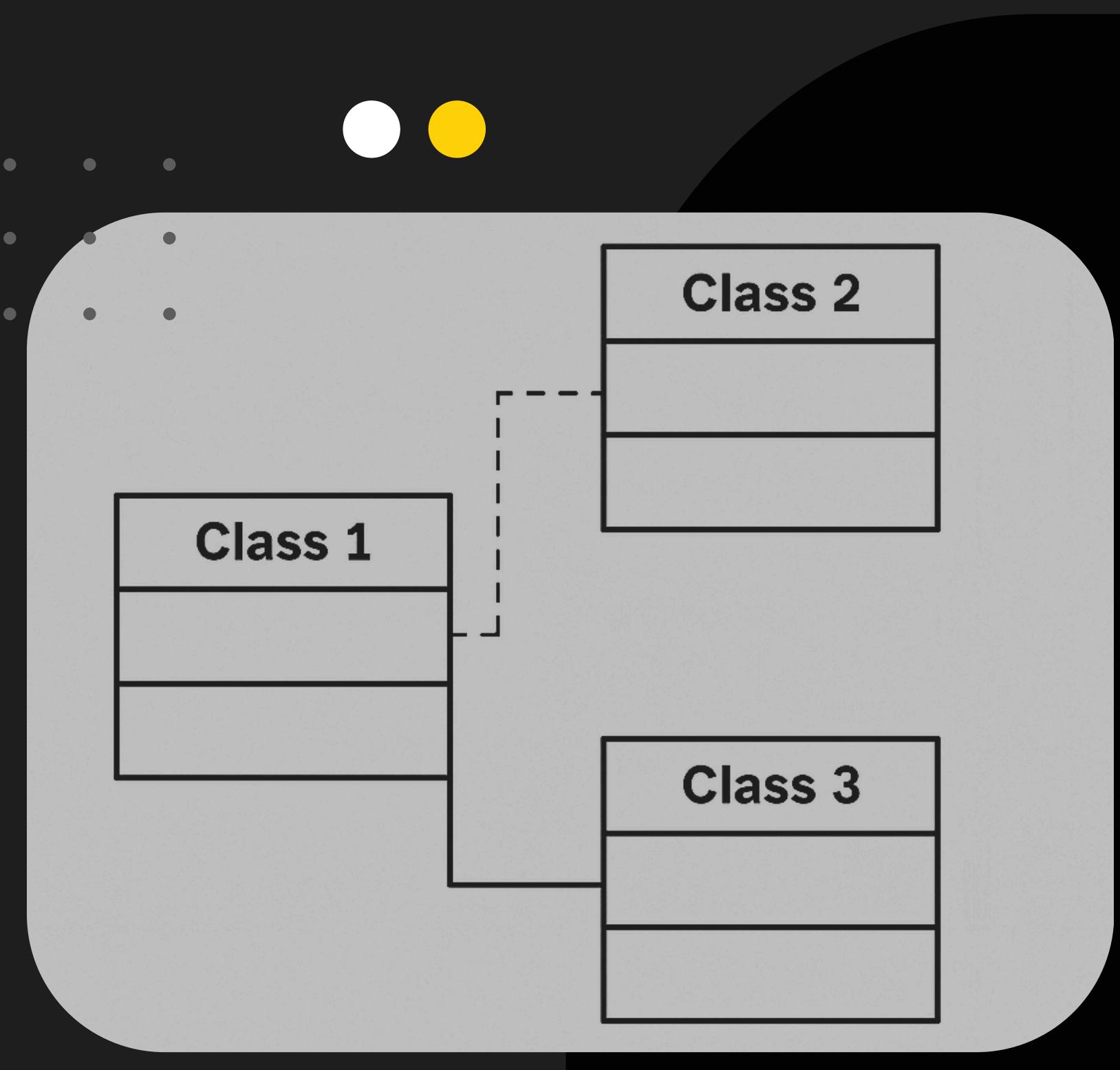




TABLA DE CONTENIDOS

01.

INTRODUCCIÓN

Introducción e
historia breve

FUNDAMENTOS DE TIPOS DE UML UML

Bases teóricas y
definiciones

03.

Descripción de los
diferentes tipos de
diagramas UML

04.

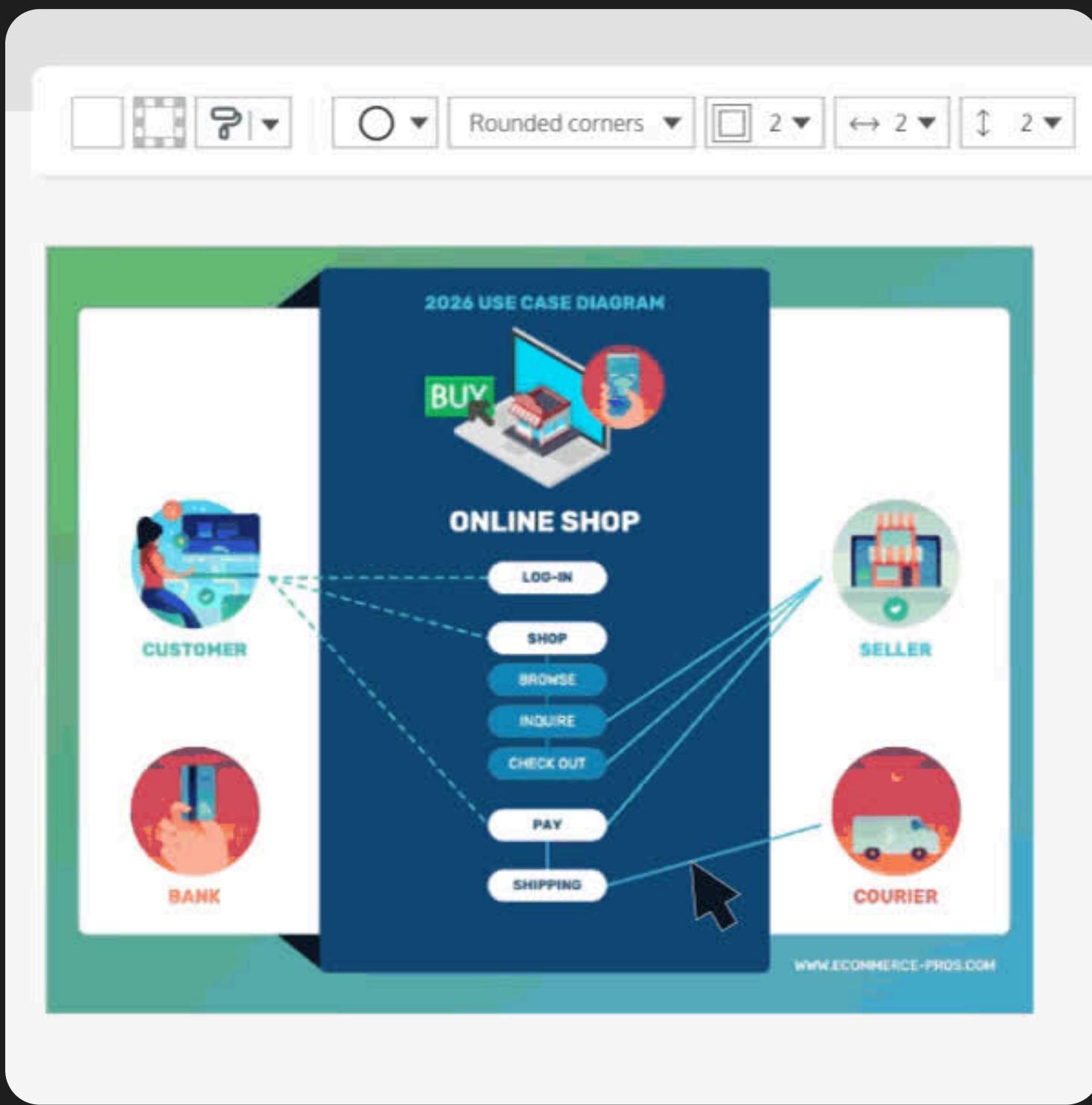
APLICACIONES

Casos de usos,
herramientas para
diagramar

05.

CONCLUSIÓN

Cierre del tema



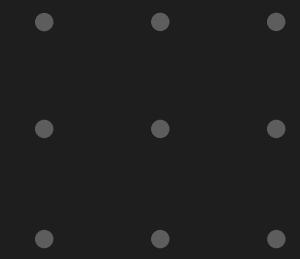
INTRODUCCIÓN

El UML fue desarrollado por Grady Booch, Ivar Jacobson y Jim Rumbaugh, la primera versión se ofreció en 1997 a OMG (Object Management Group) que gestiona estándares con la tecnología orientada a objetos. Desde aquella versión ha habido varias revisiones que gestionan dicho grupo con sus consecuentes versiones



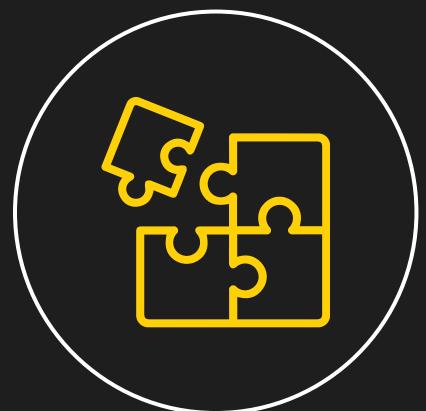


MODELADO UML



DEFINICIÓN

Es un lenguaje basado en diagramas para la especificación, visualización, construcción y documentación de cualquier sistema complejo



IMPORTANCIA

Permite comprender, comunicar y documentar sistemas complejos de manera visual y estandarizada.

Su relevancia no se limita a la programación, sino que abarca todas las etapas del desarrollo y análisis de sistemas, e incluso áreas fuera de la informática.





COMPONENTES PRINCIPALES

Elementos básicos que conforman su estructura:



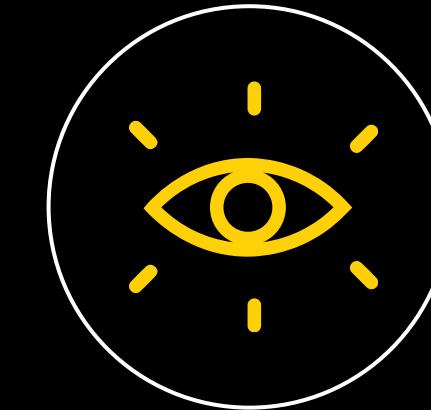


ESTRUCTURALES



Estatico

COMPORTAMIENTO



Dinamico



ESTRUCTURALES

- **Diagrama de clases:** muestra las clases, atributos, métodos y relaciones.
- **Diagrama de objetos:** ejemplos concretos de clases.
- **Diagrama de componentes:** módulos físicos de software.
- **Diagrama de despliegue:** distribución del sistema en hardware y nodos.
- **Diagrama de paquetes:** agrupación lógica de clases o componentes.



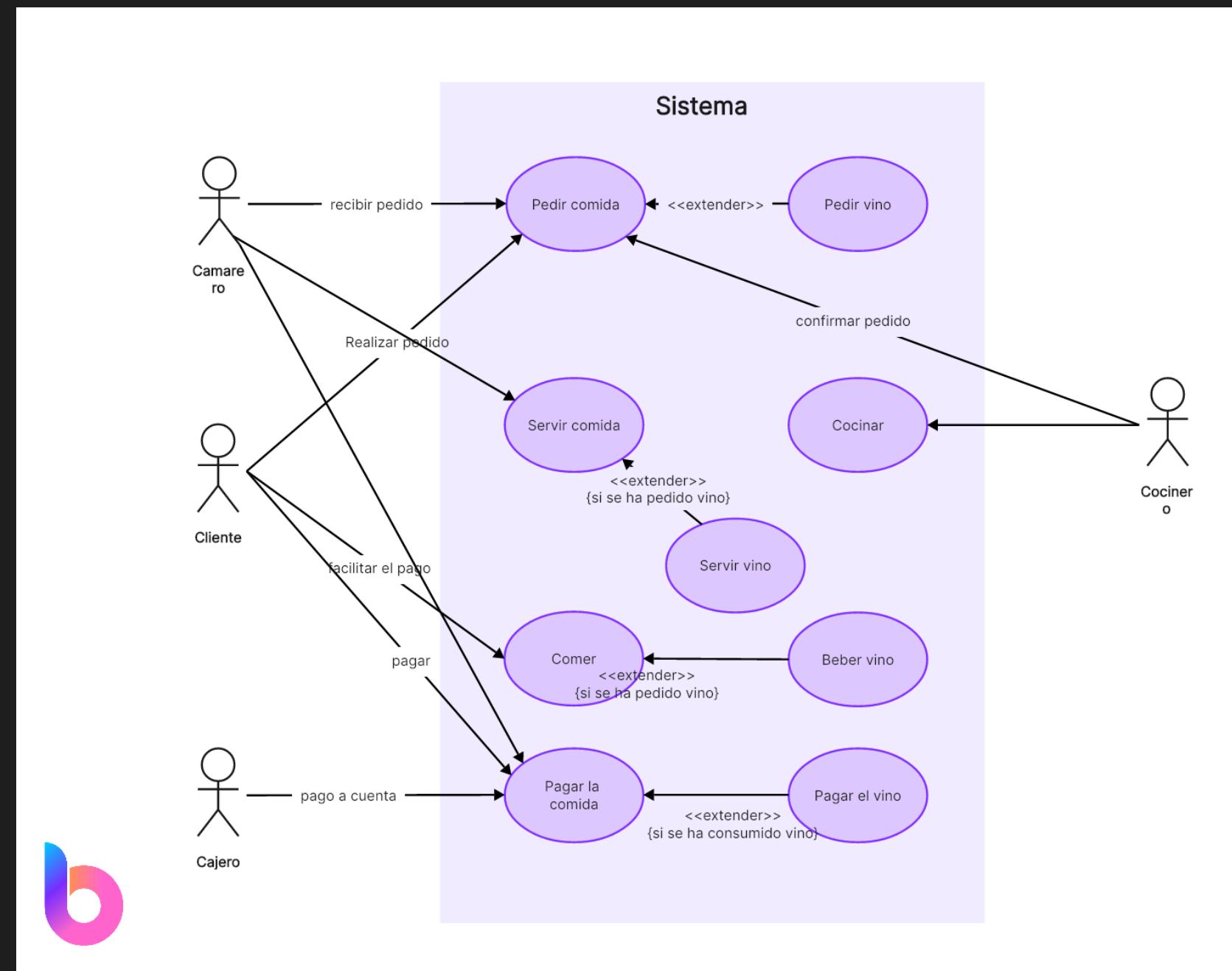
COMPORTAMIENTO

- **Casos de uso:** muestra la interacción entre actores y el sistema.
- **Diagrama de secuencia:** orden de mensajes entre objetos.
- **Diagrama de actividad:** flujo de procesos o tareas.
- **Diagrama de estado:** transición de estados de un objeto.
- **Diagrama de comunicación:** cómo se envían mensajes entre objetos.



EJEMPLO 1

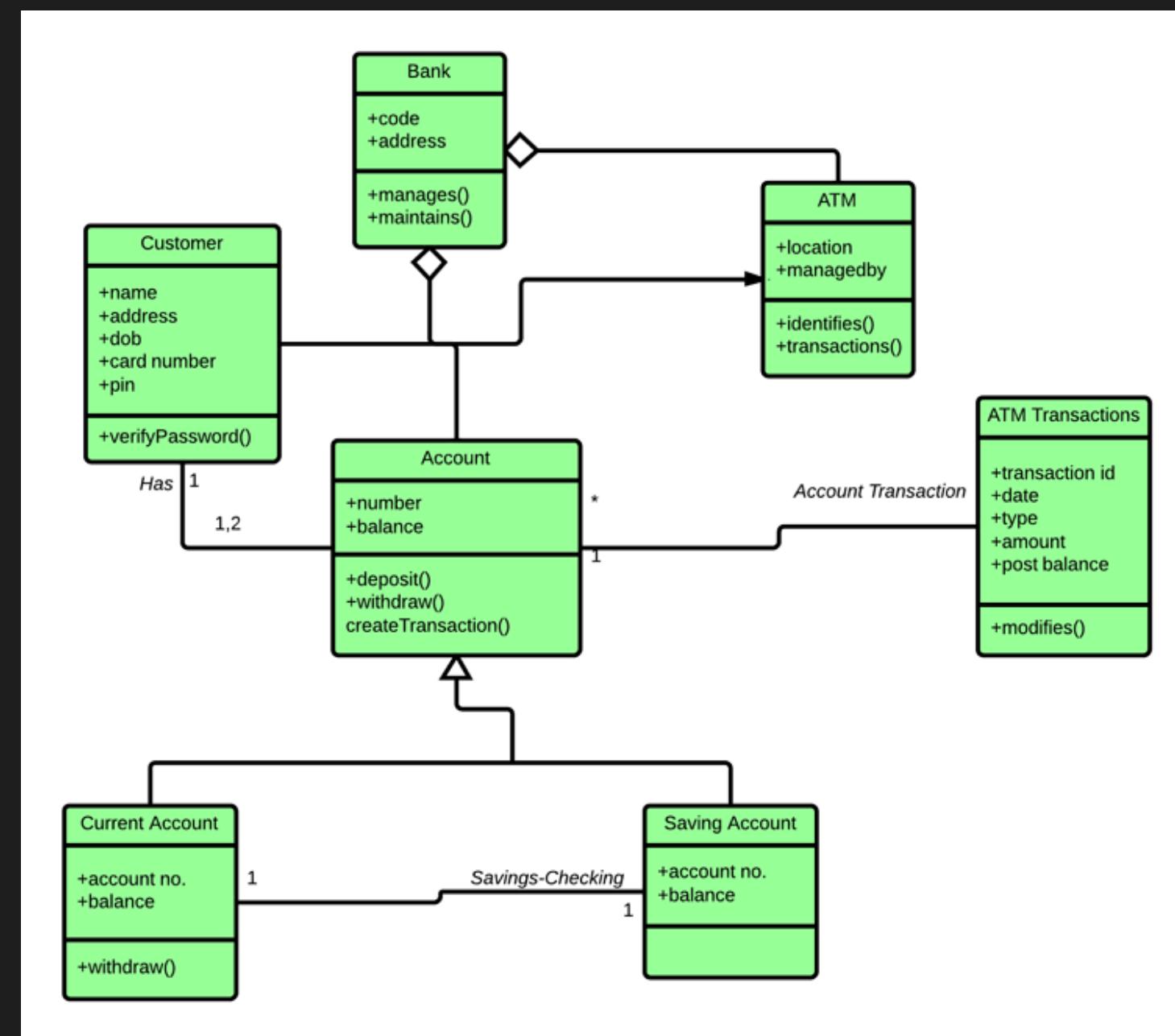
DIAGRAMA DE CASOS DE USO





EJEMPLO 2

DIAGRAMA DE CLASES



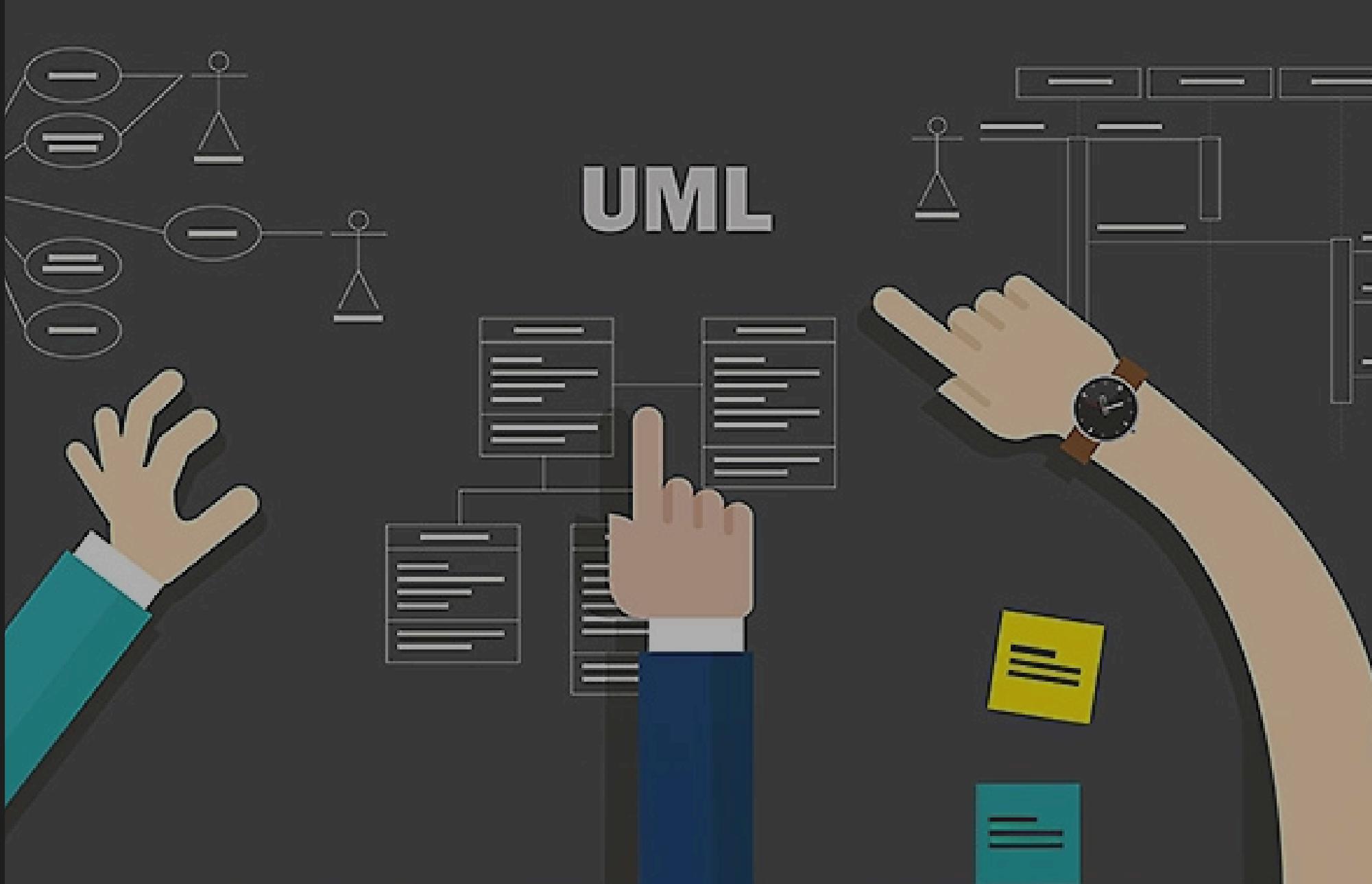


APLICACIONES DEL UML

Aplicaciones Principales:

Ingeniería de Software
Análisis de Sistemas
Modelado Empresarial

¿Por qué usar UML?



📋 Estandarización: Lenguaje común entre equipos multidisciplinarios

💬 Comunicación: Facilita el entendimiento entre técnicos y clientes

⚡ Visualización: Representación clara de sistemas complejos

🔍 Detección Temprana: Identificación de errores antes de codificar



HERRAMIENTAS PARA CREAR DIAGRAMAS UML



STARUML

Software profesional
con múltiples tipos
de diagramas



LUCIDCHART

Herramienta en línea
colaborativa y fácil
de usar



VISUAL PARADIGM

Solución completa
para modelado
empresarial



DRAW.IO

Gratuita, de código
abierto y muy
versátil



ENTERPRISE ARCHITECT

Para proyectos
empresariales de
gran escala



PLANTUML

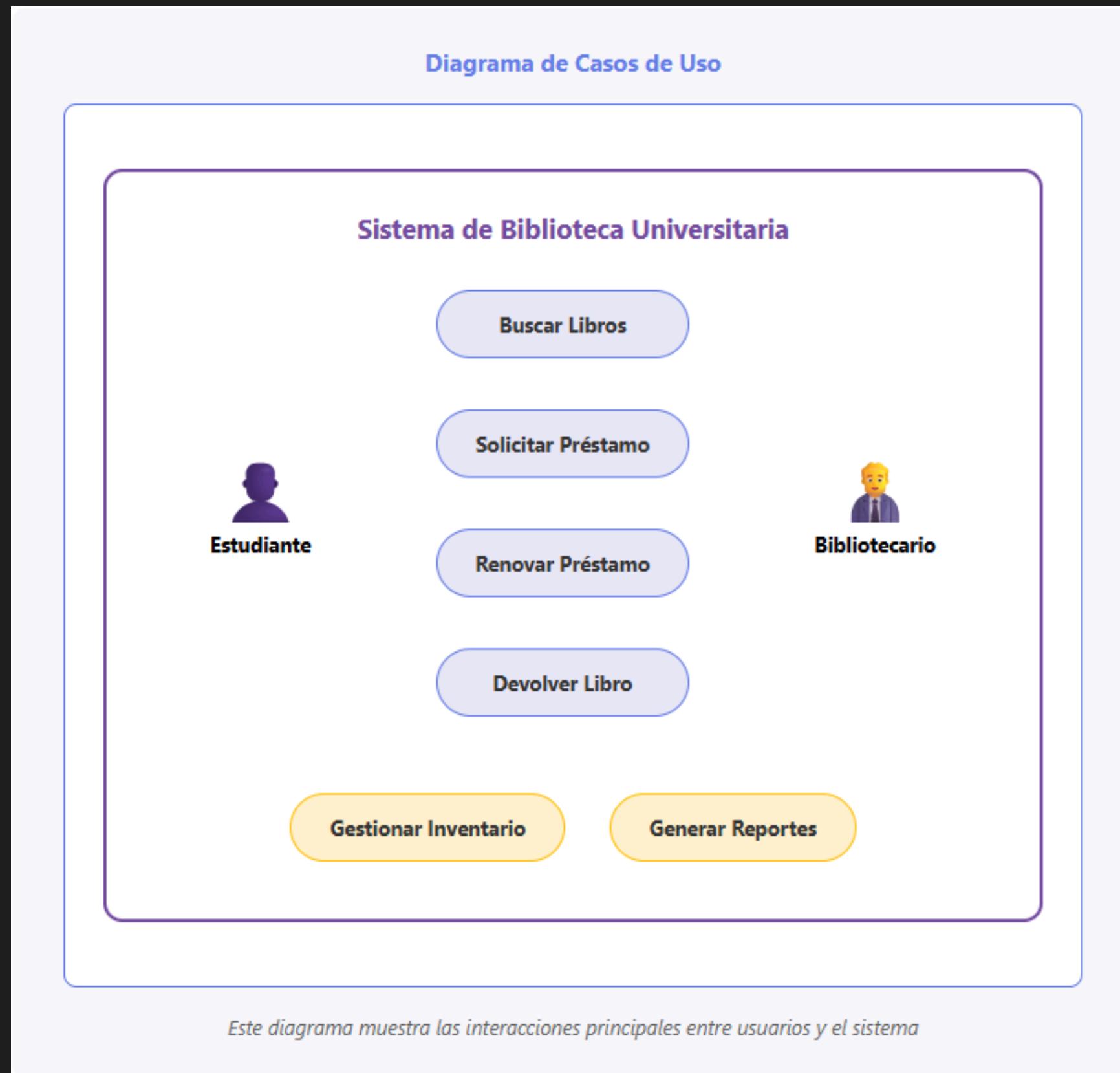
Diagramas mediante
código de texto

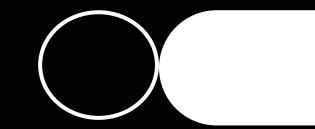
Recomendación para Estudiantes:

Comienza con Draw.io (gratuita) o Lucidchart (versión educativa). Son
intuitivas y perfectas para aprender



EJEMPLO PRÁCTICO: SISTEMA DE BIBLIOTECA





LIMITACIONES DEL UML

- **Curva de aprendizaje:** Requiere tiempo dominar todos los tipos de diagramas y su sintaxis correcta
- **Puede ser excesivo:** Modelar en exceso puede ralentizar el proyecto
- **No reemplaza la documentación ni la codificación:** Los diagramas complementan pero no sustituyen especificaciones detalladas
- **Requiere mantenimiento:** Los diagramas deben actualizarse conforme evoluciona el sistema



BUENAS PRÁCTICAS

- ✓ **Simplicidad:** Usa solo los diagramas necesarios para tu proyecto
- ✓ **Consistencia:** Mantén una nomenclatura y estilo uniforme en todos los diagramas
- ✓ **Actualización:** Mantén los diagramas sincronizados con los cambios del código
- ✓ **Colaboración:** Revisa los diagramas con el equipo antes de implementar
- ✓ **Enfoque iterativo:** Refina los modelos conforme avanza el desarrollo



CONCLUSIONES

PARA ESTUDIANTES

Desarrolla pensamiento estructurado y aprende a analizar sistemas complejos antes de codificar.

PARA EMPRESAS

Mejora la calidad del software y reduce costos al detectar problemas tempranamente.

PARA EQUIPOS

Facilita la colaboración, comunicación y el entendimiento entre todos los involucrados en un proyecto.

PARA PROYECTOS

Proporciona documentación clara que facilita el mantenimiento y evolución del sistema.

El valor del UML: es una herramienta poderosa que estandariza la comunicación en el desarrollo de software, permitiendo visualizar, diseñar y documentar sistemas de manera efectiva.

