

Manual de Introducción a UML

[Enlace a Github](#)

¿Qué es UML?

UML (Lenguaje Unificado de Modelado) es un lenguaje gráfico estandarizado para visualizar, especificar, construir y documentar los componentes de un sistema de software. UML ayuda a representar de forma visual el diseño y comportamiento de un sistema, facilitando la comunicación entre desarrolladores, analistas, clientes y otros interesados.

Fue creado para unificar y estandarizar los métodos de modelado que existían en los años 90 y desde entonces se ha convertido en el estándar más utilizado para el análisis y diseño de sistemas orientados a objetos.

¿Para qué sirve UML?

- Diseñar software antes de escribir código.
- Comprender cómo interactúan los diferentes componentes del sistema.
- Comunicar claramente la arquitectura y funcionalidades entre miembros del equipo.
- Detectar errores de diseño desde una etapa temprana.
- Documentar el comportamiento del sistema para mantenimiento futuro.

Tipos principales de diagramas UML

UML se compone de varios tipos de diagramas, que se dividen en dos grandes grupos:

1. Diagramas estructurales:

- Diagrama de clases
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de despliegue

2. Diagramas de comportamiento:

- Diagrama de casos de uso
- Diagrama de secuencia
- Diagrama de actividades
- Diagrama de estados

Diagrama de Casos de Uso

Este diagrama muestra qué funciones ofrece el sistema y quién las utiliza. Es ideal para recoger los requisitos funcionales.

Elementos clave:

- Actor: entidad externa (persona o sistema) que interactúa con el sistema.
- Caso de uso: funcionalidad que ofrece el sistema al actor.
- Sistema: contenedor que agrupa los casos de uso.

Relaciones comunes:

- Asociación: conexión entre actor y caso de uso.
- <<include>>: un caso de uso incluye a otro siempre.
- <<extend>>: un caso de uso extiende a otro opcionalmente.
- Generalización: herencia entre actores o casos de uso.

Ejemplo práctico:

Sistema: Cajero automático

Actores: Usuario, Banco

Casos de uso: Insertar tarjeta, Validar PIN, Retirar dinero, Consultar saldo

Diagrama de Secuencia

Este diagrama representa la secuencia de mensajes entre objetos para cumplir una funcionalidad.

Elementos clave:

- Objeto: identificado como nombre:Clase.
- Línea de vida: línea vertical punteada.
- Activación: rectángulo sobre la línea de vida.
- Mensaje: flecha horizontal con nombre del método.

Tipos de mensajes:

- Sincrónico: la ejecución espera respuesta (flecha llena).
- Asincrónico: no bloquea el flujo (flecha de punta vacía).
- Retorno: respuesta opcional (línea de puntos).

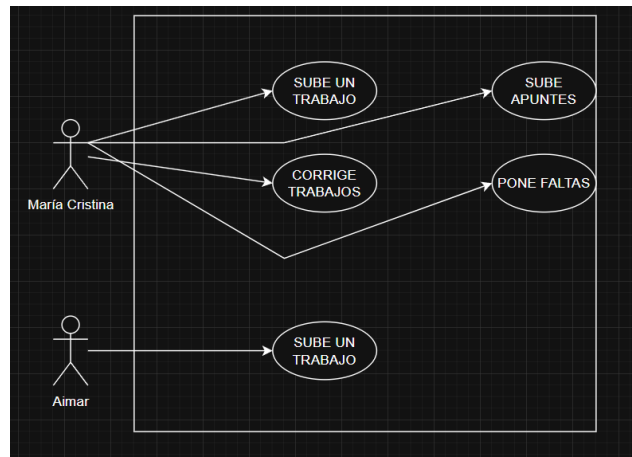
Control de flujo:

- loop: repite una secuencia.
- opt: condición opcional.
- alt: caminos alternativos.
- par: concurrencia o procesos paralelos.

Resumen práctico

Diagrama	¿Qué muestra?	¿Para qué sirve?
Casos de uso	Qué puede hacer el sistema	Capturar requisitos
Secuencia	En qué orden ocurren los eventos	Comprender interacciones
Clases	Estructura del sistema	Modelar la arquitectura
Actividad	Flujo de procesos	Describir tareas y decisiones

Diagrama de casos de uso



Este es un **diagrama de casos de uso UML** que representa las interacciones entre dos actores (María Cristina y Aimar) y un sistema.

Actores:

- **María Cristina:** Tiene múltiples responsabilidades dentro del sistema.
- **Aimar:** Tiene una única interacción con el sistema.

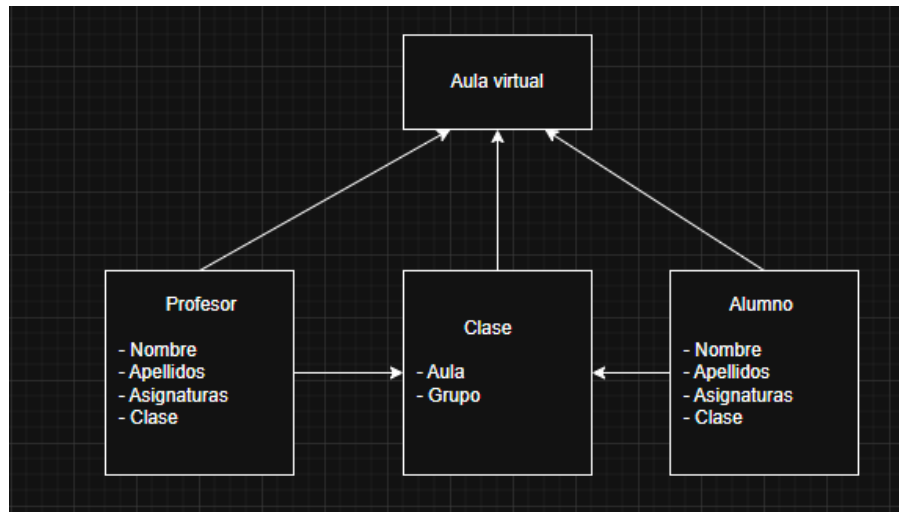
Casos de uso:

- **SUBE UN TRABAJO:** Acción que realizan ambos actores.
- **SUBE APUNTES:** Relacionado únicamente con María Cristina.
- **CORRIGE TRABAJOS:** Solo accesible para María Cristina.
- **PONE FALTAS:** Selecciona los alumnos que no vienen a clase.

Relaciones:

- Las flechas indican **asociación** entre los actores y los casos de uso.
- Existen **relaciones entre casos de uso**, lo cual puede interpretarse como **<<include>>**.

Diagrama de clases



Este diagrama de clases modela la estructura básica de un **aula virtual** con tres clases principales:

Aula Virtual

- Representa la plataforma o sistema global en el que ocurren todas las interacciones.
- Se conecta a la clase **Clase** mediante una relación de dependencia (flecha punteada hacia “Clase”), lo que indica que el aula virtual gestiona o contiene a las instancias de “Clase”.

Clase

Contiene los atributos:

- Aula (identificador o ubicación virtual del aula)
- Grupo (identificador del grupo o sección dentro de la clase)

Recibe flechas desde **Profesor** y **Alumno**, lo que indica una asociación bidireccional:

- Un **Profesor** imparte una **Clase**.
- Un **Alumno** está inscrito en una **Clase**.

Profesor

- Atributos:
 - Nombre
 - Apellidos
 - Asignaturas (lista de materias que imparte)
 - Clase (referencia a la clase que dirige)
- Flecha hacia **Clase**: un profesor “apunta” o asigna su clase en el aula virtual.

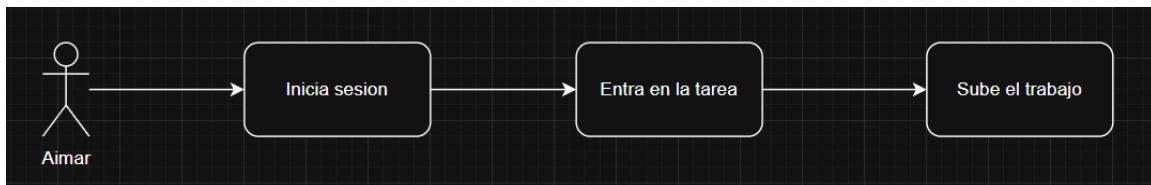
Alumno

- Atributos:
 - Nombre
 - Apellidos
 - Asignaturas (lista de materias en las que está matriculado)
 - Clase (referencia a la clase a la que pertenece)
- Flecha hacia **Clase**: un alumno “se inscribe” en una clase dentro del aula virtual.

Relaciones clave

- **Aula Virtual → Clase**: dependencia, el sistema depende de las clases que gestiona.
- **Profesor ↔ Clase**: asociación, un profesor imparte una o varias clases, y cada clase es impartida por al menos un profesor.
- **Alumno ↔ Clase**: asociación, un alumno asiste a una o varias clases, y cada clase tiene múltiples alumnos.

Diagrama de actividades o de secuencia



Actor:

- **Aimar:** Representado por el "muñequito", es el usuario que interactúa con el sistema.

Flujo de Actividades:

1. **Inicia sesión:**
Aimar accede a la plataforma ingresando sus credenciales. Esta es la primera acción que inicia el flujo.
2. **Entra en la tarea:**
Una vez dentro, navega hacia la sección o módulo donde se encuentra la tarea asignada.
3. **Sube el trabajo:**
Finalmente, Aimar carga o entrega su trabajo (puede ser un archivo o una respuesta) en la plataforma.

Observaciones:

- El flujo es **lineal y secuencial**, no hay decisiones ni bifurcaciones.
- Es un buen ejemplo de un proceso de tipo "caso de uso" desde la perspectiva del usuario.
- Técnicamente también podría representarse con un **diagrama de caso de uso** o **diagrama de secuencia**, si se quisiera ver más detalle de la interacción con el sistema.