

# Tema 4: Lenguajes formales para modelos conceptuales

## MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS Curso 2025-2026

## B1 - Sistemas Dinámicos

**T1:** Fundamentos del modelado de sistemas dinámicos

**T2:** Estabilidad, controlabilidad y observabilidad

**T3:** Paradigmas de simulación

## B2 - Sistemas Complejos

**T4:** Lenguajes formales para modelos conceptuales

**T5:** Redes complejas y modelado estructural

## B3 - Modelado con IA

**T6:** Identificación de sistemas

**T7:** Inteligencia artificial aplicada al modelado de sistemas

1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
5. Modelado basado en agentes

## 1. Introducción

2. UML para modelado de sistemas

3. SysML para modelado de sistemas físicos y ciber-físicos

4. Formalismo DEVS

5. Modelado basado en agentes

# 1. Introducción

- El **modelado de sistemas** complejos requiere herramientas que permitan expresar la estructura y el comportamiento de los sistemas de manera precisa, **sin ambigüedades**.
- Los lenguajes formales cumplen esta función, al **proporcionar una sintaxis y semántica** bien definidas que posibilitan la comunicación rigurosa entre l@s diseñador@s, ingenier@s y simuladores.
- Estos lenguajes se utilizan para describir la **estructura**, el **comportamiento** y las **restricciones** de un sistema.
- Entre los más utilizados se encuentran **UML**, **SysML** y **DEVS**, además de los lenguajes específicos de dominio (**DSLs**) y los enfoques de modelado basado en agentes.

# 1. Introducción

Un lenguaje formal permite definir de manera estructurada:

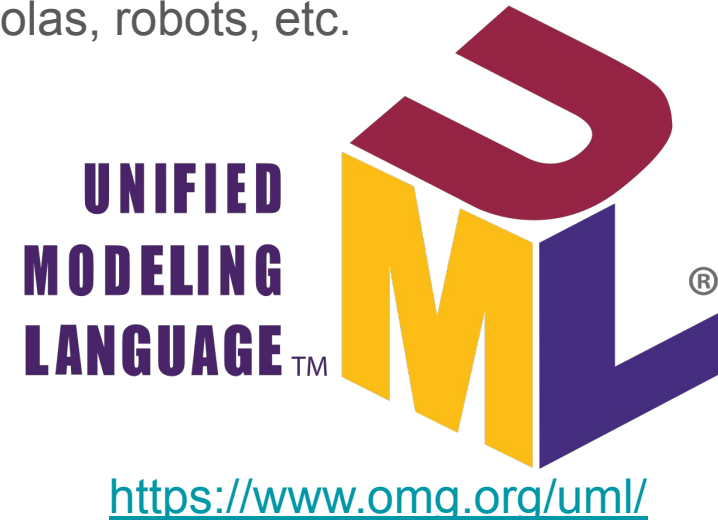
- Elementos de modelado: entidades, clases, bloques, agentes o componentes.
  - Relaciones: asociación, composición, flujo o dependencia.
  - Restricciones: condiciones algebraicas o lógicas sobre el sistema.
- El uso de diagramas facilita la visualización de las diferentes **vistas** de un sistema: estructural, de comportamiento, de requisitos y física.

1. Introducción
- 2. UML para modelado de sistemas**
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
5. Modelado basado en agentes

## 2. UML para modelado de sistemas

UML (**Unified Modeling Language**) es un **lenguaje gráfico semi-formal**. Tiene una sintaxis bastante rígida (cuadros, flechas, tipos de relaciones ...) pero su semántica puede adaptarse al dominio: software, colas, robots, etc.

- **Describir la estructura** del sistema (qué entidades hay y cómo se relacionan).
- **Describir el comportamiento** (cómo cambia con el tiempo, qué eventos hay).
- Servir como puente entre el **modelo conceptual** y la **implementación / simulación**.





## 2. UML para modelado de sistemas

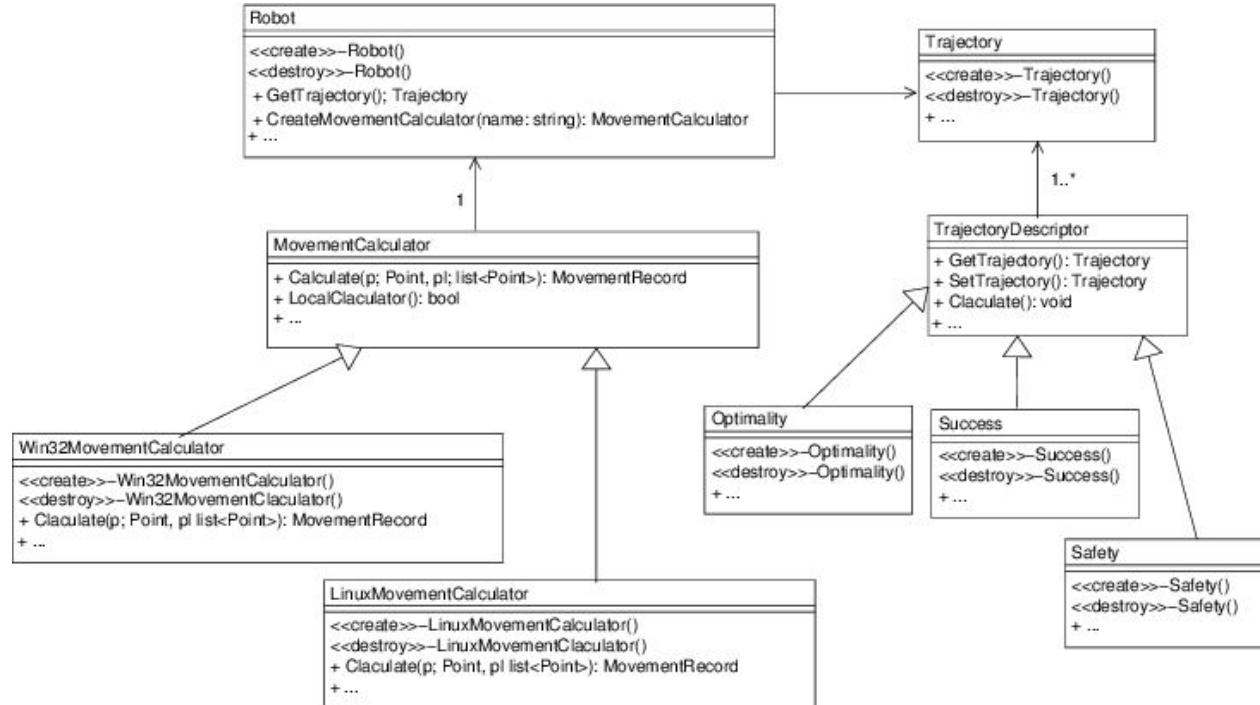
- **Clases** (rectángulos):
  - Nombre de la clase (parte superior).
  - Atributos (parte central).
  - Operaciones/métodos (parte inferior).
- **Atributos:** nombre : Tipo
  - Opcionalmente, visibilidad: + público, - privado, # protegido.
- **Operaciones:** operación(parámetros) : TipoRetorno

## 2. UML para modelado de sistemas

### Relaciones:

- **Asociación:** línea simple entre clases. Puede tener:
  - *Multiplicidad* (1, 0..1, \*, 1..\*) en los extremos.
  - *Rol* (nombre del papel: `cliente`, `servidor`, etc.).
- **Agregación:** rombo blanco → “tiene-un” pero débil.
- **Composición:** rombo negro → “forma-parte-de” fuerte.
- **Herencia (generalización):** triángulo blanco hacia la superclase.
- **Dependencia:** línea discontinua con flecha, indica uso de una clase por otra.

## 2. UML para modelado de sistemas



1. Introducción
2. UML para modelado de sistemas
- 3. SysML para modelado de sistemas físicos y ciber-físicos**
4. Formalismo DEVS
5. Modelado basado en agentes

### 3. SysML para modelado de sistemas físicos



UNIVERSITAT D'ALACANT  
UNIVERSIDAD DE ALICANTE  
Escola Politècnica Superior  
Escuela Politécnica Superior

El lenguaje SysML (**S**ystems **M**odeling **L**anguage) es una extensión y especialización de UML diseñada para la **ingeniería de sistemas**. Su propósito es integrar en un mismo modelo los aspectos estructurales, funcionales, físicos y de requisitos de sistemas que combinan **hardware, software, control, comunicación y procesos humanos**.

Mientras que UML está orientado a la ingeniería de software, SysML amplía sus capacidades para representar:

- Relaciones entre subsistemas mecánicos, eléctricos, electrónicos y de control.
- Restricciones físicas y ecuaciones de comportamiento dinámico.
- Requisitos del sistema y su trazabilidad hacia componentes concretos.
- Integración de hardware y software dentro de un mismo modelo conceptual.



<https://sysml.org/>

### 3. SysML para modelado de sistemas físicos



UNIVERSITAT D'ALACANT  
UNIVERSIDAD DE ALICANTE  
Escola Politècnica Superior  
Escuela Politécnica Superior

Los modelos SysML son muy utilizados en el diseño de:

- **Sistemas ciber-físicos**, como robots, vehículos autónomos, drones, etc.
- **Sistemas embebidos** con sensores, actuadores y software de control.
- **Sistemas industriales** (plantas de producción, líneas automatizadas).
- **Sistemas aeroespaciales**

### 3. SysML para modelado de sistemas físicos



UNIVERSITAT D'ALACANT  
UNIVERSIDAD DE ALICANTE  
Escola Politècnica Superior  
Escuela Politécnica Superior

Elementos y tipos de diagramas en SysML:

SysML conserva la estructura modular de UML, pero introduce nuevos tipos de diagramas que permiten una descripción multidisciplinar del sistema. Los diagramas más relevantes para la simulación de sistemas son los siguientes:

1. Block Definition Diagram (BDD)
2. Internal Block Diagram (IBD)
3. Requirement Diagram
4. Parametric Diagram
5. Activity y State Machine Diagrams

### 3. SysML para modelado de sistemas físicos



UNIVERSITAT D'ALACANT  
UNIVERSIDAD DE ALICANTE  
Escola Politècnica Superior  
Escuela Politécnica Superior

1. **BDD:** bloques Robot, Controlador, Sensor, Motor, Bateria, Entorno.
2. **IBD:** conexiones internas entre sensores, controlador y motores.
3. **Requirement diagram:** requisitos R1–R3 (velocidad, detección, autonomía).
4. **Parametric diagram:** ecuaciones dinámicas para calcular movimiento y consumo de energía.
5. **State Machine:** comportamiento del robot (parado, avanzando, evitando).



### 3. SysML para modelado de sistemas físicos



UNIVERSITAT D'ALACANT  
UNIVERSIDAD DE ALICANTE  
Escola Politècnica Superior  
Escuela Politécnica Superior

#### Ventajas de SysML

- Permite modelar **estructuras, requisitos y comportamiento** en un mismo lenguaje.
- Facilita la integración de diferentes disciplinas (mecánica, eléctrica, software, control).
- Es compatible con herramientas de simulación (por ejemplo, exportación a Modelica).
- Favorece la **trazabilidad**: los requisitos se relacionan directamente con los bloques del diseño.

En resumen, SysML constituye un lenguaje unificado que conecta el **modelo conceptual**, las **restricciones físicas** y el **comportamiento dinámico**, facilitando la simulación integral de sistemas ciber-físicos complejos.

1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
- 4. Formalismo DEVS**
5. Modelado basado en agentes

## 4. Formalismo DEVS

El formalismo DEVS (**Discrete Event System Specification**) es un marco matemático para modelar y simular **sistemas de eventos discretos**. Proporciona una definición precisa de los componentes del sistema (modelos atómicos y acoplados) y de su comportamiento en el tiempo a través de **transiciones de estado** provocadas por eventos.

La gran ventaja de DEVS es que:

- Separa claramente la **especificación del modelo** de la **mecánica de la simulación**.
- Permite composicionalidad: modelos simples se pueden combinar en sistemas complejos.
- Tiene una semántica formal bien definida, lo que facilita el análisis.

**TUTORIAL:** <https://hosting.cs.vt.edu/hpcs2008/DEVSTutorial.pdf>

## 4. Formalismo DEVS

### 4.1. Modelo atómico DEVS

Un **modelo atómico** DEVS describe el comportamiento de un componente elemental de un sistema de eventos discretos. Se define como una tupla:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

## 4. Formalismo DEVS

- $X$ : conjunto de **entradas** (eventos de entrada). Cada entrada se asocia a un puerto y un valor.
- $S$ : conjunto de **estados internos** del modelo.
- $Y$ : conjunto de **salidas** (eventos de salida).
- $\delta_{int} : S \rightarrow S$ : **función de transición interna**. Describe cómo cambia el estado cuando vence el tiempo de permanencia del estado actual (sin eventos externos).

## 4. Formalismo DEVS

- $\delta_{ext} : S \times X \times \mathbb{R}^+ \rightarrow S$ : **función de transición externa**. Describe cómo cambia el estado cuando llega un evento externo, antes de que venza el tiempo interno. El parámetro de tiempo indica cuánto tiempo ha transcurrido desde la última transición (interna o externa).
- $\lambda : S \rightarrow Y$ : **función de salida**. Determina la salida generada justo antes de que se produzca una transición interna.
- $ta : S \rightarrow \mathbb{R}_0^+$ : **función de tiempo de avance** (*time advance*). Indica cuánto tiempo permanecerá el sistema en el estado actual si no hay eventos externos.

## 4. Formalismo DEVS

### 4.3. Modelo acoplado DEVS

Los **modelos acoplados** (o *coupled models*) permiten construir sistemas complejos combinando varios modelos atómicos y/o acoplados. Un modelo acoplado DEVS se define como:

$$N = \langle X_N, Y_N, D, \{M_d\}_{d \in D}, EIC, EOC, IC \rangle$$

## 4. Formalismo DEVS

- $X_N$ : conjunto de **entradas** del modelo acoplado.
- $Y_N$ : conjunto de **salidas** del modelo acoplado.
- $D$ : conjunto de **componentes** (índices de los submodelos).
- $M_d$ : modelo DEVS (atómico o acoplado) para cada  $d \in D$ .



## 4. Formalismo DEVS

- *EIC (External Input Coupling)*: acoplamiento de las entradas externas  $X_N$  con las entradas de los submodelos.
- *EOC (External Output Coupling)*: acoplamiento de las salidas de los submodelos con las salidas externas  $Y_N$ .
- *IC (Internal Coupling)*: acoplamiento interno entre las salidas de unos submodelos y las entradas de otros.

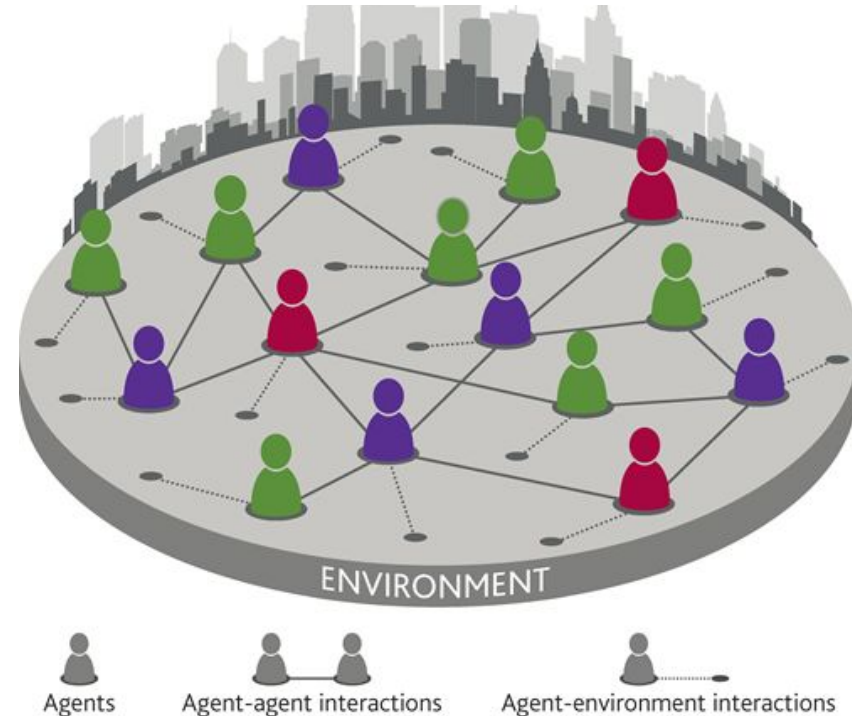
1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
- 5. Modelado basado en agentes**

## 5. Modelado basado en agentes

Un modelo basado en agentes se compone de tres elementos principales:

- **Agentes:** entidades autónomas que poseen un conjunto de estados internos, reglas de comportamiento y capacidad para percibir y actuar sobre su entorno.
- **Entorno:** espacio (físico o abstracto) en el que los agentes interactúan, que puede contener recursos, obstáculos o información compartida.
- **Interacciones:** mecanismos mediante los cuales los agentes comunican información o modifican su entorno.

$$A_i = \langle S_i, P_i, R_i, f_i \rangle$$



## 5. Modelado basado en agentes

- $S_i$ : conjunto de estados internos del agente  $i$ .
- $P_i$ : conjunto de percepciones que el agente obtiene del entorno o de otros agentes.
- $R_i$ : conjunto de reglas o políticas de decisión.
- $f_i$ : función de comportamiento que define cómo el agente cambia de estado o actúa en respuesta a  $P_i$ .

El sistema completo se describe como un conjunto de agentes:

$$\mathcal{A} = \{A_1, A_2, \dots, A_N\}$$

que interactúan dentro de un entorno  $E$  definido por sus propias dinámicas.

# Tema 4: Lenguajes formales para modelos conceptuales

## MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS Curso 2025-2026