

El Diagrama de Estados en UML

En todo momento, un objeto en un sistema orientado a objetos se encuentra en un estado determinado. El diagrama de estados en UML se utiliza precisamente para modelar el comportamiento dinámico de dichos objetos o incluso de casos de uso completos. Esta herramienta permite representar cómo un objeto cambia de estado en respuesta a determinados eventos, los cuales generalmente son generados por agentes externos, ya sea el usuario, otros sistemas o componentes del mismo sistema. El diagrama especifica qué eventos provocan transiciones entre estados y qué respuestas o acciones se ejecutan como consecuencia de estas transiciones.

La notación utilizada para representar estas transiciones sigue los estándares establecidos por UML y permite describir de manera precisa el paso de un objeto desde un estado inicial a uno final, a través de diversas situaciones intermedias. Para entender bien este tipo de diagramas es fundamental conocer sus elementos estructurales, entre los que se encuentra el concepto de vértice. Un vértice es una abstracción de un nodo dentro del grafo de estados, es decir, representa un punto desde el cual o hacia el cual puede dirigirse una transición. Existen dos tipos de vértices: los estados propiamente dichos y los pseudoestados.

El estado, como vértice principal del diagrama, modela una situación en la que las condiciones del objeto permanecen estables o invariantes durante un tiempo determinado. Esto no significa que el objeto esté inactivo, sino que, dentro de ese estado, puede estar realizando actividades mientras sus condiciones esenciales no cambian. Por ejemplo, un estado puede representar que una impresora está “imprimiendo”. En UML se reconocen tres tipos de estados: el estado simple, el estado compuesto y el estado submáquina de estados.

El estado simple es el más básico. No contiene subestados ni regiones ni tampoco está compuesto por otras máquinas de estados. Se representa mediante un rectángulo de bordes redondeados, que suele etiquetarse con un verbo en gerundio escrito en negrita. Este estado puede incluir varias actividades y transiciones internas. Algunas de las actividades asociadas a un estado simple incluyen las que se ejecutan al entrar (entry), las que tienen lugar durante el estado (do), y las que se desencadenan al salir del estado (exit). También pueden definirse eventos de entrada o condiciones de guarda, así como eventos de salida que se activan cuando el estado concluye. Todos estos elementos ayudan a especificar el comportamiento interno del estado de forma detallada.

El estado compuesto es más avanzado, ya que contiene dentro de sí otros subestados, formando una jerarquía. Estos subestados pueden ser secuenciales o disjuntos, cuando pertenecen a una única región, o concurrentes, también llamados ortogonales, cuando se distribuyen en dos o más regiones paralelas. Cada región puede tener su propio estado inicial y final. El estado compuesto permite representar comportamientos más

complejos sin perder claridad, y se etiqueta por fuera de su contenedor para facilitar su identificación.

El tercer tipo de estado es la submáquina de estados. Este tipo de estado tiene un significado semántico similar al estado compuesto, ya que encapsula una lógica más compleja en su interior. La diferencia es que una submáquina de estados representa un resumen de una máquina de estados más detallada, que se desarrolla en otro diagrama aparte. Este enfoque modular facilita la organización y reutilización del diseño.

Además de los estados propiamente dichos, en los diagramas de estados también se utilizan pseudoestados. Estos son vértices abstractos que no representan estados en sí mismos, sino que sirven para conectar distintas partes del diagrama o para indicar puntos de decisión y control del flujo. Entre los pseudoestados más comunes se encuentran el pseudoestado inicial, que marca el comienzo del comportamiento del objeto y se representa gráficamente como un círculo negro; y el pseudoestado final, que indica el final de la ejecución y suele representarse con un círculo con un borde blanco grueso.

Otros pseudoestados incluyen el punto de entrada y el punto de salida, que se utilizan en estados compuestos y submáquinas para controlar cómo se accede o se abandona la lógica interna de esos estados. También existen pseudoestados de selección, que permiten evaluar condiciones de guarda para decidir por qué transición continuar; de unión o join, que agrupan múltiples transiciones provenientes de diferentes regiones ortogonales en una única transición de salida; y de bifurcación o fork, que permiten dividir una transición en varias que se dirigen a regiones ortogonales distintas. Existe además un pseudoestado de unión general, que puede actuar como join o fork según el contexto. Finalmente, los pseudoestados de historia, tanto superficial (shallow history, representado con una H) como profunda (deep history, representado con una H*), permiten recordar el subestado más recientemente activo o incluso toda la configuración interna de subestados de un estado compuesto.

Gracias a esta rica variedad de elementos, el diagrama de estados en UML es capaz de representar con gran precisión y flexibilidad la evolución de los objetos dentro de un sistema a medida que estos responden a estímulos externos. Esta herramienta resulta indispensable en el diseño de sistemas complejos, especialmente cuando se requiere modelar comportamientos dependientes del tiempo, de eventos o de condiciones específicas del entorno.