[**Canva**](https://www.canva.com/design/DAFJ_4FWLDQ/fjdTDDmXGBFapZ9duQgq4w/edit?utm_content=DAFJ_4FWLDQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

**¿Qué es un diagrama de estado UML?**

Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) es uno de los 14 tipos de diagrama definidos en el lenguaje de modelado unificado (UML), o Unified Modeling Language, y en el lenguaje de modelo de sistemas (SysML, del inglés Systems Model Language). Otros tipos de diagramas UML son, por ejemplo, el diagrama de casos de uso y el diagrama de componentes.

Esta herramienta muestra los estados que tendrá un objeto durante su ciclo de vida. Es decir, es un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. El diagrama proporciona un estado inicial, uno final y al menos un estado intermedio para cada objeto. A su vez permite de este modo, representar el ciclo de vida completo de cualquier sistema, subsistema, componentes o clases del mismo.

Dichos sean: Una puerta mecánica, un lector de libros electrónicos o un componente tecnológico de un vehículo, entre otros.

presentación:

Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) es uno de los 14 tipos de diagrama definidos en el lenguaje de modelado unificado (UML).

Esta herramienta muestra los estados que tendrá un objeto durante su ciclo de vida. Es decir, es un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados.

**¿Para qué sirve?**

Sirve para representar gráficamente los estados, eventos, acciones y transiciones de los objetos y dar una idea más completa de su funcionamiento.

Por más que se basen en unos pocos elementos, combinarlos de manera inteligente nos permite representar fácilmente secuencias de estado complejas.

**¿En qué se podría utilizar?**

Suelen emplearse, por ejemplo, en el diseño de sistemas embebidos (en inglés, embedded systems), donde las señales automatizadas y los procesos en segundo plano deben estar perfectamente coordinados. Sin embargo, tiene una alta utilidad para las siguientes tareas:

Representar objetos basados en eventos en un sistema reactivo.

Ilustrar escenarios de casos de uso en un contexto de negocios.

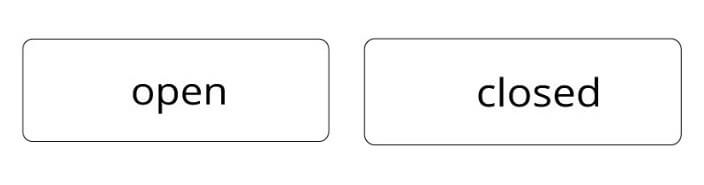
Describir cómo se mueve un objeto a través de diversos estados a lo largo de su existencia.

Mostrar el comportamiento general de una máquina de estados o el comportamiento de un conjunto relacionado de máquinas de estados.

**Estados**

Un estado identifica una condición o una situación en la vida de un objeto durante la cual satisface alguna condición, ejecuta alguna actividad o espera que suceda algún evento. Un objeto permanece en un estado durante un tiempo finito (no instantáneo).

Un estado se representa gráficamente por medio de un rectángulo con los bordes redondeados.



Los dos posibles estados de una puerta: puede estar abierta o cerrada, pero no ambas cosas al mismo tiempo.

En este caso en específico la puerta está condicionada a encontrarse en uno de los dos estados, pues no existe uno intermedio. Debido a esta restricción, no puede estar abierta y cerrada al mismo tiempo.

En los diagramas de estado más elaborados, el rectángulo puede dividirse en hasta tres zonas donde se muestran especificaciones de comportamiento. Estas zonas son divisiones internas las cuales alojan el nombre del estado, el valor característico de los atributos del objeto en ese estado y las acciones que se realizan en ese estado, respectivamente.

En muchos diagramas se suelen omitir los dos compartimentos inferiores debido a la complejidad que estos representan.

**Transición:**

Una transición es el proceso en el cual un objeto cambia de un estado a otro mediante la intervención de un evento.

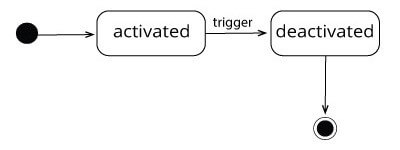
Esta transición de estado comunica los estados entre sí y se representa mediante una flecha. Puede haber condiciones para que se desencadene dicha transición. En términos generales, los diagramas de estado UML representan transiciones internas y externas. Un diagrama de estado siempre debe presentar alguna transición externa, pues hace falta para poder alcanzar el otro estado, pero no es obligatorio que incluya transiciones internas.

En el diagrama de estado de una puerta de supermercado, por ejemplo, se podría especificar la siguiente condición para la acción “cerrar la puerta del supermercado”: que la puerta del supermercado haya estado abierto al menos cinco segundos sin la actividad del sensor antes de que el estado cambie de “abierto” a “cerrado”.

#### Transición externa: cambio de estado

La transición que figura en el siguiente ejemplo se considera externa y tiene como resultado que el objeto cambie de estado (entry/exit).

Ejemplo: después de que se active la alarma de una radio-despertador, el estado cambia de “alarma activada” a “alarma desactivada”.



Cuando se activa la alarma, el objeto cambia de estado: si hace un momento la alarma estaba activada, ahora está desactivada.

#### Transición interna: estado inalterado

Una transición interna no desencadena un cambio de estado, sino una actividad.

Ejemplo: algunos sistemas de contabilidad vuelven a enviar las facturas sin pagar automáticamente al cliente (transición externa). Si lo que envían es un recordatorio de que la factura está pendiente de abonar, este representa una transición interna: es decir, aunque hay una actividad (“enviar el recordatorio”), la factura permanece en el mismo estado (“no pagada”) hasta nuevo aviso.

### Eventos: ¿por qué se cambia de estado?

Mediante los eventos es posible describir con más detalle las condiciones bajo las cuales se abandona un estado para pasar al siguiente. En el caso de la transición del estado inicial al primer estado real, no es necesario, pero se puede añadir más información de manera opcional.

Si no se indica ningún evento, significa que el evento ocurre automáticamente tan pronto como se hayan finalizado todas las actividades en los estados anteriores.

Si no se indica el desencadenante, significa que este evento siempre está teniendo lugar. Los eventos pueden representarse como una especificación de comportamiento dentro del estado o dentro de la transición hacia otro estado (ver transición).

Un evento desencadenante debe cumplir las siguientes tres condiciones:

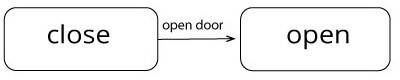
* entry: el evento se activa automáticamente cuando se desencadena un estado, es decir, en todas las transiciones entrantes.
* exit: el evento se desencadena cuando se abandona un estado, es decir, en todas las transiciones salientes.
* do: el evento se desencadena una y otra vez si no se cambia de estado.

Estas indicaciones pueden anotarse dentro del propio estado para simplificar la representación del comportamiento bajo el cual se cambia de estado. Hay dos opciones para mostrar estos desencadenantes de manera gráfica. Una de ellas es indicarlos dentro del recuadro de estado correspondiente, como ilustra el siguiente ejemplo de diagrama de estado:



El estado de una puerta es “cerrada”. Para entrar en este estado, primero debe tener lugar el evento “cerrar la puerta” (entry). Cuando se abandona el estado, se produce el evento “abrir la puerta” (exit). Durante el estado, “la puerta está (permanentemente) cerrada” (do).

Los eventos también se pueden indicar mediante una flecha de transición:



En los diagramas de estado sencillos, los eventos se anotan sobre la flecha de transición.

### Pseudoestados

En los diagramas de estado UML, si algún elemento de control influye en el funcionamiento de una máquina de estados, pero no tiene asignado ningún valor, se denomina pseudo estado. En UML 2, la versión actual del lenguaje de modelado unificado, se definen los siguientes diez pseudoestados:

* Estado inicial (en inglés, initial): sin transición entrante y con una transición saliente que revela cuál es el estado al principio de la secuencia.
* Estado final (en inglés, final): sin transición saliente; fin de la secuencia de comportamiento.
* Bifurcación (en inglés, fork): división en varios estados paralelos.
* Sincronización (en inglés, join): sincronización de varios estados paralelos.
* Unión (en inglés, junction): nodo de unión de varias transiciones en serie.
* Elección (en inglés, choice): nodo desde el cual pueden iniciarse diversas transiciones sobre la base de una decisión previa.
* Punto de entrada (en inglés, entrypoint): síntesis de transiciones similares que entran en un estado compuesto.
* Punto de salida (en inglés, exitpoint): síntesis de transiciones similares que se originan en un estado compuesto.
* Historial superficial (en inglés, shallowhistory): almacenamiento del último subestado activo de un estado compuesto.
* Historial profundo (en inglés, deephistory): almacenamiento del último subestado activo de todos los niveles jerárquicos de un estado compuesto.

## Diagramas complejos

Dependiendo de la complejidad del proceso, es posible incluir subestados en el esquema que muestra una imagen detallada de cada estado del objeto y de su posible comportamiento. Esta versión más compleja de los diagramas de estado UML suele ser la más habitual, especialmente en el ámbito técnico.

* Estado compuesto: esta estructura permite definir un estado en profundidad.

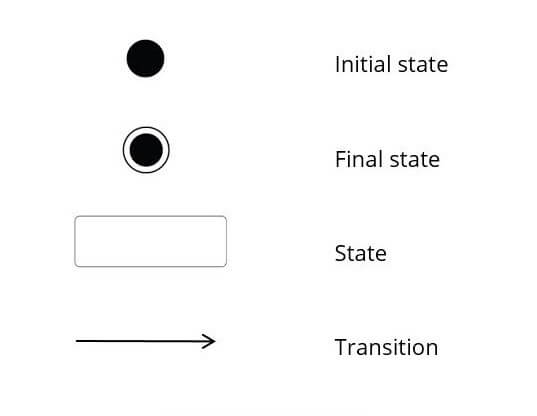
Ejemplo: una puerta puede estar en dos estados: “abierta” o “cerrada”. Los subestados del estado “cerrada” podrían ser “bloqueada” y “desbloqueada”.

* Estado de submáquina: el estado incluye un diagrama de estado subordinado. Un subestado que consista en varios subestados se denomina un estado complejo. Los diversos subestados pueden tanto ejecutarse independientemente el uno del otro como estar relacionados entre sí.

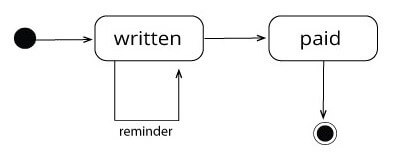
Ejemplo: la función de despertador de un smartphone es una de las muchas funciones que puede estar relacionada con otros estados. Si se programan distintas alarmas para diferentes horas y días de la semana, el proceso completo consistirá en subestados que se ejecutan de forma independiente.

Crear un diagrama de estado: ejemplo de un diagrama simple

Los diagramas de estado pueden aplicarse a objetos de todo tipo. En el siguiente ejemplo, te mostramos cómo incluir cada elemento en el diagrama de una factura. Estos son los elementos más importantes de un diagrama de estado UML:

Los elementos más importantes de un diagrama de estado UML.

Si combinamos los elementos para nuestro ejemplo, un diagrama sencillo podría tener este aspecto:

En este ejemplo, el diagrama de estado UML presenta una transición interna.

En el punto de partida, la factura se encuentra en el pseudoestado “escrita” (written). En el mejor de los casos, se producirá la transición hacia el estado “pagada” (paid). Esta transición podría describirse con más detalle indicando el evento “enviar”.

Una vez que se ha pagado, la factura se encuentra en el estado “pagada”. Antes de llegar a este estado, puede ser necesario enviar un “recordatorio” (reminder). En este caso, como la factura no cambia de estado, aunque se ocasione una actividad, se trata de una transición interna. Si no se paga la factura, se enviarán hasta tres recordatorios.

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-estado-uml/>

<http://www.vc.ehu.es/jiwotvim/IngenieriaSoftware/Teoria/BloqueII/UML-5.pdf>

<https://www.lucidchart.com/pages/es/diagrama-de-maquina-de-estados>