

Modelado con Máquinas de Estados

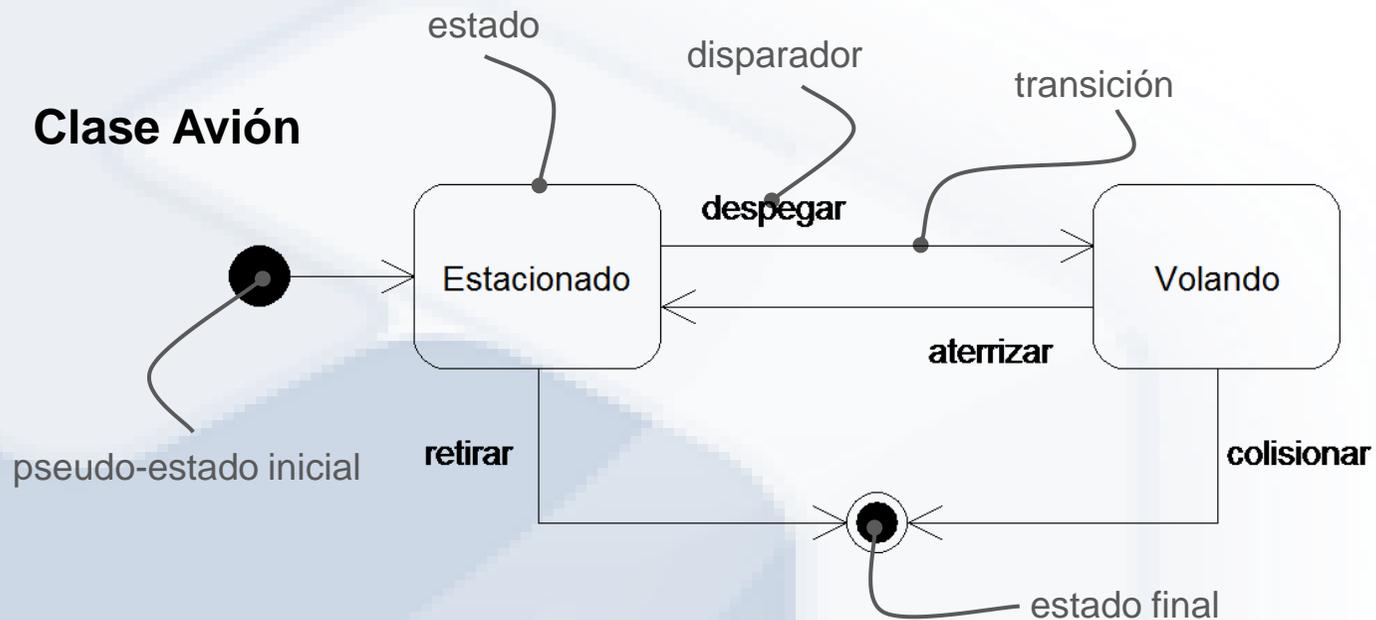


Máquinas de Estados

- Modelado dinámico: dos tipos complementarios de diagramas.
 - Diagramas de interacción: **ilustran** o ejemplifican el comportamiento, aunque también permiten especificar la dinámica de una sociedad particular de objetos.
 - Diagramas estados: **especifican** el comportamiento de modo abstracto.
- Una máquina de estados es una descripción **abstracta** del comportamiento.
 - Suele estar asociada a una o varias clases.
 - Describe el comportamiento de los objetos de una clase, en respuesta a eventos externos.
 - Contempla todas las situaciones y reacciones relevantes.
 - Típicamente es cíclica.
- Los objetos de una clase con máquina de estados asociada tienen **memoria**.
 - Esperan la llegada de **eventos** externos (típicamente mensajes).
 - **Reaccionan** de forma distinta en función del estado que recuerdan.
 - **Determinismo**: un mismo evento no puede conducir a estados distintos.
- El emisor de los mensajes no se representa en un diagrama de estados.
 - No es necesario conocer el **origen de los eventos** para describir la reacción,
 - pero es muy útil conocerlo para comprender el comportamiento global del sistema.

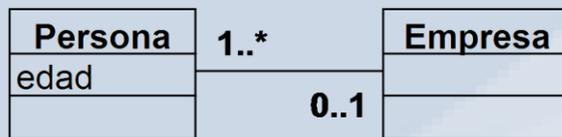
Estados, Disparadores y Transiciones

- Los **disparadores** causan las **transiciones** entre **estados**

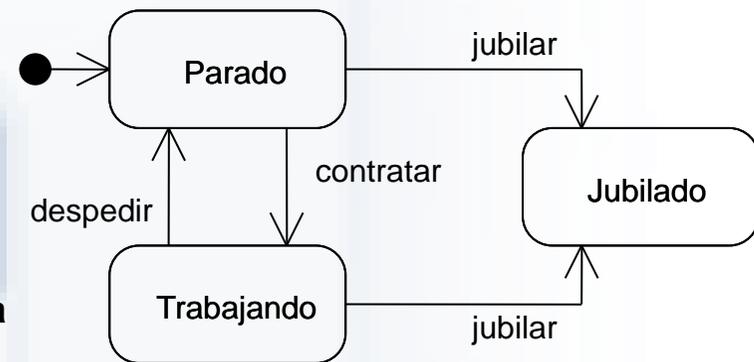


Estados y Transiciones

- Un estado es una situación en la vida de un objeto caracterizada por **satisfacer** una condición: **esperar** un evento (estática) o **realizar** una actividad (dinámica).
- Un estado tiene **duración**, una transición es **instantánea**.
- Cada estado tiene un **nombre** (sustantivo, participio, gerundio...).
- El estado de un objeto está relacionado con los **valores** de sus atributos, los **enlaces** con otros objetos y las **actividades** que esté realizando.
- Estados distintos implican **reacciones distintas** ante el mismo evento.
- No obstante, los **estados semánticamente relevantes** de una clase no pueden determinarse automáticamente (ej.: ¿Un estado por cada valor de Persona.edad?).

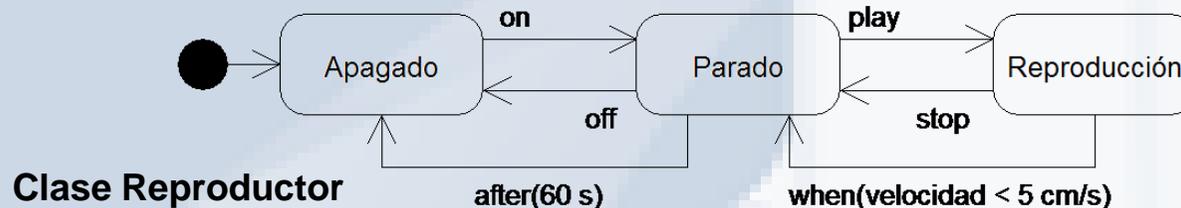


Clase Persona



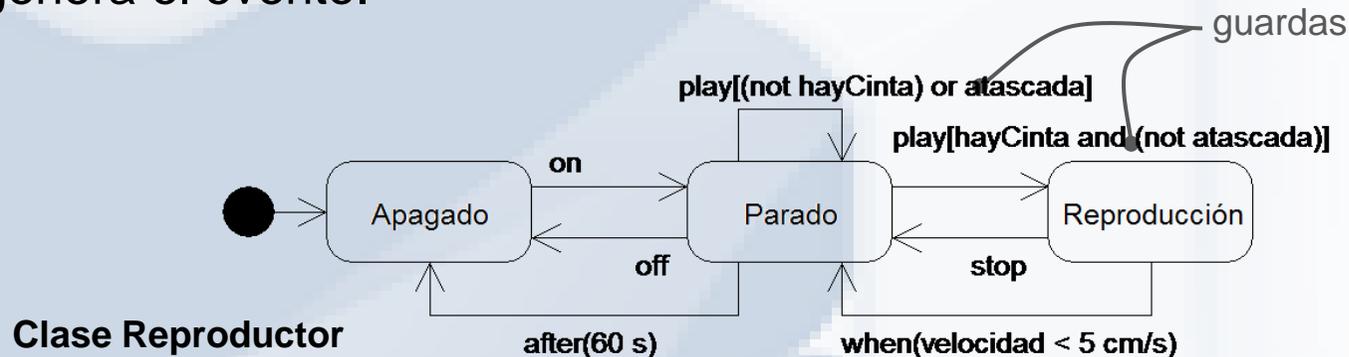
Disparadores, Eventos y Transiciones

- Un evento representa la **ocurrencia de un suceso**, dentro o fuera del objeto, que provoca un cambio de estado en el objeto (**dispara una transición**).
 - La sucesión de transiciones marca el “camino” seguido por el objeto entre los estados.
 - Estados y transiciones representan respectivamente **intervalos e instantes** de tiempo.
 - Una transición puede tener varios eventos vinculados.
- **Tipos de eventos:**
 - Evento de **llamada**. Recepción de mensaje síncrono: **operación**.
 - Evento de **señal**. Recepción de mensaje asíncrono: **señal**.
 - Evento de **cambio**. Comprobación continua de una condición lógica: **when()**.
 - Evento **temporal**. Tiempo transcurrido desde la entrada en el estado: **after()**.
- Los mensajes recibidos, síncronos o asíncronos, pueden tener **parámetros**.



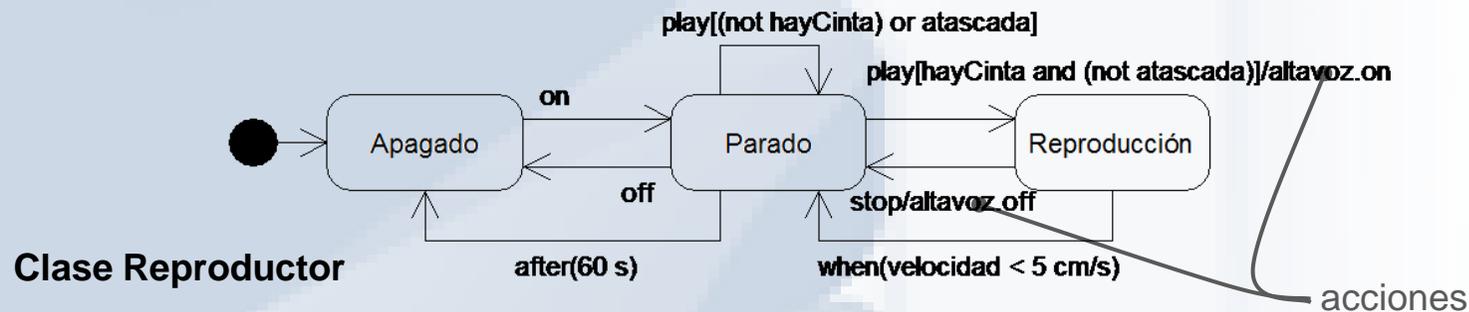
Transiciones con Guardas

- Una guarda es una condición lógica que es **evaluada** en el momento de recibir el evento y **autoriza** o no la transición de estado (guarda tras evento).
 - Si la transición no es autorizada por la guarda, el evento **no tiene efecto**.
 - Varias transiciones disparadas por un mismo evento pueden estar guardadas por **condiciones mutuamente exclusivas**.
 - La condición lógica debe ser evaluable (expresión lógica) en el contexto de la clase propietaria (**parámetros** del evento, **atributos** y **enlaces** del objeto, etc.).
- No confundir guarda con **evento de cambio**: la guarda requiere la ocurrencia de un evento para ser evaluada, mientras que la condición asociada al evento de cambio es evaluada continuamente y ella misma genera el evento.



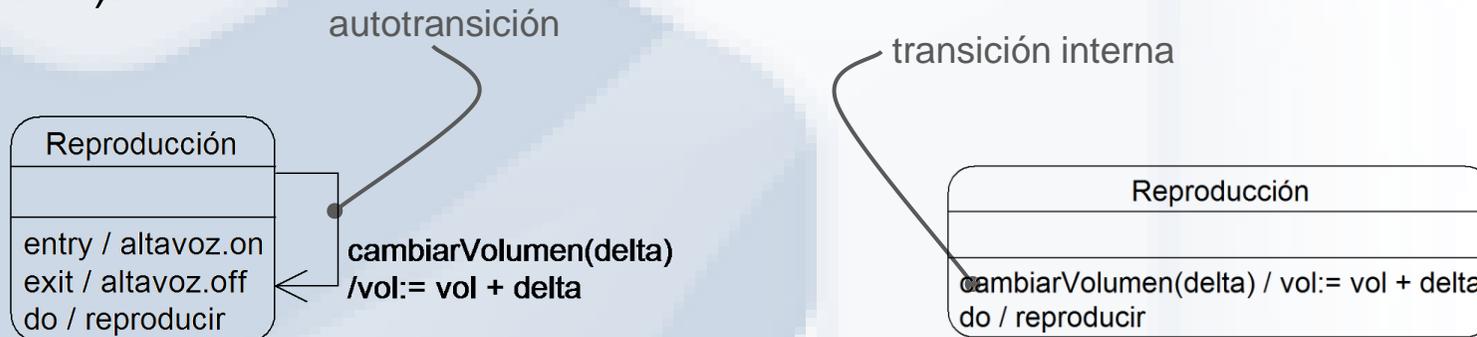
Transiciones con Acciones

- Una acción es un **comportamiento** que se ejecuta de modo **instantáneo** (con tiempo de ejecución despreciable) y **atómico** (no interrumpible).
 - Está asociada a un **evento** que dispara una transición.
 - Es **completada** antes de que la transición alcance el nuevo estado.
 - Puede definirse recursivamente como **secuencia de acciones**.
- Efectos
 - **En el propio objeto**: modificar atributos o enlaces, invocar operaciones locales.
 - **En otros objetos**: mensajes síncronos (operaciones) o asíncronos (señales).



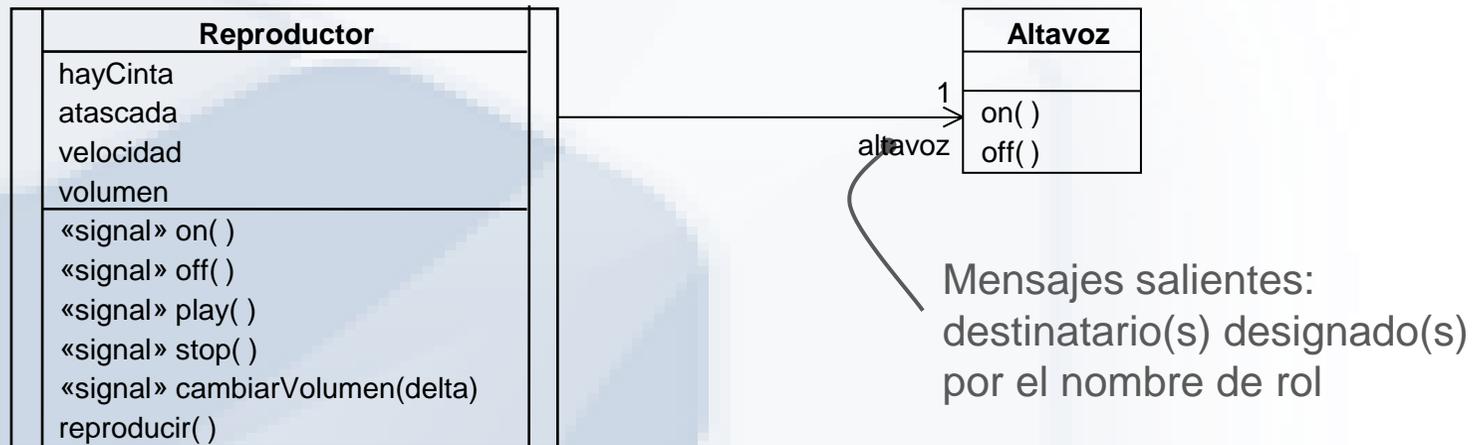
Estados con Acciones y Actividades

- Dentro de un estado pueden ejecutarse dos tipos de comportamientos:
 - **Acciones** asociadas a eventos puntuales que ocurren dentro de un estado.
 - Acción de entrada: **entry**
 - Acción de salida: **exit**
 - Acción por evento interno: **evento**
 - **Actividades** asociadas al estado como tal: **do**
 - Una actividad es **duradera** e **interrumpible**.
 - Puede modificar una condición que genere un **evento de cambio**.
- No confundir transición interna con **autotransición** (sale y entra en el estado).



Diagramas de Estados vs. Diagramas de Clases

- **Eventos:** operaciones o señales de la clase receptora.
- **Guardas** y eventos de cambio: valores de atributos (y asociaciones).
- **Acciones:** trozos de operaciones.
- **Actividades:** operaciones de la clase receptora.
- Tanto clases activas como pasivas pueden tener máquinas de estados.



¿Qué Ocurre Cuando se Recibe un Evento?

1. ¿Es **relevante** para el estado actual?
 - Si no hay ninguna transición o evento interno disparados por ese evento, el evento se descarta.
2. ¿Existe una **guarda** asociada al evento?
 - Si la transición no es autorizada por la guarda, el evento no tiene efecto.
3. ¿Existe una **actividad** asociada al estado origen?
 - Se para.
4. ¿Existe una **acción de salida**?
 - Se ejecuta la acción o secuencia de acciones, en el propio objeto o en otros.
5. ¿Existe una **acción** asociada al evento?
 - Se ejecuta la acción o secuencia de acciones, en el propio objeto o en otros.
6. ¿Existe una **acción de entrada**?
 - Se ejecuta la acción o secuencia de acciones, en el propio objeto o en otros.
7. ¿Existe una **actividad** asociada al estado destino?
 - Empieza a ejecutarse.